



C o m m u n i t y   E x p e r i e n c e   D i s t i l l e d

# phpList 2 E-mail Campaign Manager

Get to grips with the phpList e-mail announcement  
delivery system!

David Young

**[PACKT]** open source\*  
PUBLISHING community experience distilled

[www.allitebooks.com](http://www.allitebooks.com)

# phpList 2 E-mail Campaign Manager

Get to grips with the phpList e-mail announcement delivery system!

**David Young**

**[PACKT]** open source   
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

# phpList 2 E-mail Campaign Manager

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2011

Production Reference: 1180711

Published by Packt Publishing Ltd.  
32 Lincoln Road  
Olton  
Birmingham, B27 6PA, UK.

ISBN 978-1-849511-04-9

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Asher Wishkerman ([a.wishkerman@mpic.de](mailto:a.wishkerman@mpic.de))

# Credits

**Author**

David Young

**Project Coordinator**

Srimoyee Ghoshal

**Reviewer**

Deepak Vohra

**Proofreader**

Bernadette Watkins

**Acquisition Editor**

Usha Iyer

**Indexers**

Monica Ajmera Mehta

Rekha Nair

**Development Editor**

Alina Lewis

**Production Coordinator**

Melwyn D'sa

**Technical Editor**

Sakina Kaydawala

**Cover Work**

Melwyn D'sa

**Copy Editor**

Leonard D'Silva

# About the Author

**David Young**'s first experience in documenting the open source world was the official 2003 manual for JAJC, a popular jabber client, which he authored in XML using Docbook.

He followed this with the development of "Bandersnatch", the whimsically named Jabber message logger, which remains the de facto open source platform for recording and archiving messages on a Jabber server.

After founding and successfully running an open source consulting company ("Funky Penguin") for several years, David "retired" to full-time employment.

David now works for a respected New Zealand IT consulting firm (Prophecy Networks Ltd) and pursues the odd "Funky Penguin" project in his spare time.

David's projects, notes, and ramblings can be found at:  
<http://www.funkypenguin.co.nz>.

---

I'd like to thank my wife for her support, encouragement, and constant supply of Griffin's Chocolate Fingers. The team at Packt, for the opportunity to author this book. My Funky Penguin phpList clients (you know who you are), for the opportunities you provide me with. My son Zachary, of whom I'm forever a proud and loving father.

---

# About the Reviewer

**Deepak Vohra** is a consultant and a principal member of the NuBean.com software company. Deepak is a Sun Certified Java Programmer and Web Component Developer, and has worked in the fields of XML, Java programming and J2EE for over five years. Deepak is the co-author of the *Apress* book *Pro XML Development with Java Technology* and was the technical reviewer for the *O'Reilly* book *WebLogic: The Definitive Guide*. Deepak was also the technical reviewer for the Course Technology PTR book *Ruby Programming for the Absolute Beginner* and the technical editor for the *Manning Publications* book *Prototype and Scriptaculous in Action*. Deepak is also the author of the *Packt Publishing* books *JDBC 4.0* and *Oracle JDeveloper for J2EE Development*, and *Processing XML Documents with Oracle JDeveloper 11g*.

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy & paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

|  |           |
|--|-----------|
| <b>Preface</b>                                   | <b>1</b>  |
| <hr/>  |           |
| <b>Chapter 1: Installation and Configuration</b> | <b>7</b>  |
| <hr/>  |           |
| <b>Installing phpList files and database</b>     | <b>7</b>  |
| Requirements                                     | 8         |
| Downloading phpList                              | 8         |
| Unpacking and uploading phpList                  | 9         |
| Creating a database                              | 10        |
| Performing initial configuration                 | 10        |
| Database settings                                | 10        |
| Path settings                                    | 10        |
| Performing web-based configuration               | 11        |
| Initialize database                              | 11        |
| Initial login                                    | 13        |
| Change Admin Password                            | 14        |
| Configure General Values                         | 16        |
| Configure Attributes                             | 17        |
| Create Lists                                     | 19        |
| Create Subscribe Pages                           | 20        |
| <b>Testing e-mail delivery</b>                   | <b>20</b> |
| Create user                                      | 20        |
| Sending a message                                | 22        |
| Processing the message queue (in test mode)      | 24        |
| Disabling test mode                              | 24        |
| Processing the message queue (for real)          | 25        |
| Summary  | 25        |
| <hr/>  |           |
| <b>Chapter 2: Setting up Subscriber Forms</b>    | <b>27</b> |
| <hr/>  |           |
| <b>Basic concepts</b>                            | <b>27</b> |
| <b>Setting up a basic subscribe page</b>         | <b>28</b> |
| <b>Creating a custom subscribe page</b>          | <b>30</b> |



|  |           |
|--|-----------|
| Creating attributes  | 30        |
| Creating subscribe page  | 32        |
| Customizing title and text   | 33        |
| Customizing the HTML header and footer                                 | 33        |
| Customizing delivery options   | 34        |
| Selecting attributes   | 34        |
| Selecting list   | 35        |
| Protecting your subscribe page from spammers                           | 36        |
| Signing up for reCAPTCHA keys and downloading the PHP library          | 36        |
| Modifying index.php  | 36        |
| Modifying admin/subscribelib2.php                                      | 37        |
| <b>Putting a subscribe form on other applications</b>                  | <b>38</b> |
| Generating HTML code for the subscribe form                            | 38        |
| Customizing HTML code  | 39        |
| Changing the form action   | 39        |
| Removing the JavaScript  | 39        |
| Optional pop-up confirmation   | 39        |
| Removing the subscribe form button                                     | 40        |
| <b>Summary</b>   | <b>41</b> |
| <b>Chapter 3: Setting up E-mail Bounce Handling</b>                    | <b>43</b> |
| <b>E-mail bounce handling</b>  | <b>43</b> |
| Requirements   | 43        |
| Configuring basic bounce settings                                      | 44        |
| <b>Testing and manually processing bounces</b>                         | <b>46</b> |
| <b>Reviewing bounces</b>   | <b>47</b> |
| Taking action on individual bounces                                    | 48        |
| Interpreting why an e-mail bounced                                     | 49        |
| Examining users whose e-mail bounces                                   | 50        |
| Reviewing unconfirmation actions in the eventlog                       | 51        |
| <b>Summary</b>   | <b>51</b> |
| <b>Chapter 4: Setting up phpList with Popular Publishing Platforms</b> | <b>53</b> |
| <b>Integrating phpList with Drupal</b>                                 | <b>53</b> |
| Prerequisites  | 54        |
| Installing and configuring the phpList integration module              | 54        |
| External phpList configuration   | 55        |
| Attribute mapping  | 57        |
| My Account – My Newsletters options                                    | 58        |
| Registration Page options  | 59        |
| Miscellaneous  | 60        |
| Synchronize users to phpList   | 60        |
| Turn on debugging  | 61        |
| Configuring list access  | 61        |
| Configuring Drupal permissions   | 62        |

---

|  |            |
|--|------------|
| Clearing Drupal's cache                                      | 63         |
| Confirming access to My newsletters                          | 64         |
| Adding the phpList subscribe block                           | 65         |
| Troubleshooting the phpList subscribe block                  | 68         |
| <b>Integrating phpList with WordPress</b>                    | <b>68</b>  |
| Prerequisites  | 68         |
| Installing and configuring the phpList Integration plugin    | 68         |
| General Settings   | 70         |
| Form Settings  | 70         |
| Adding a phpList Integration page                            | 71         |
| Adding a phpList Integration widget                          | 72         |
| <b>Integrating phpList with Facebook</b>                     | <b>74</b>  |
| Prerequisites  | 74         |
| Preparing phpList  | 75         |
| Creating the Facebook app                                    | 77         |
| Configuring the Facebook page                                | 80         |
| Making phpList the default tab on your Facebook page         | 82         |
| <b>Summary</b>   | <b>83</b>  |
| <b>Chapter 5: List, User, and Administrator Management</b>   | <b>85</b>  |
| <b>Individual user management</b>                            | <b>86</b>  |
| Updating a user's details and list memberships (as the user) | 87         |
| Unsubscribing a user (as the user)                           | 88         |
| Examining a user's history (as administrator)                | 88         |
| Updating a user's details (as admin)                         | 89         |
| Bulk user management   | 90         |
| Importing users  | 90         |
| Importing a simple list                                      | 91         |
| Importing a complex list                                     | 93         |
| Reconciling users  | 97         |
| Exporting users  | 100        |
| All users  | 100        |
| Specific list  | 101        |
| <b>Managing lists</b>  | <b>102</b> |
| <b>Creating a list</b>                                       | <b>102</b> |
| Performing member operations on a list                       | 103        |
| <b>Managing administrators</b>                               | <b>105</b> |
| Creating a restricted admin                                  | 105        |
| Setting admin permissions                                    | 107        |
| Creating a super admin                                       | 109        |
| Assigning administrators to lists                            | 109        |
| Testing a new administrator                                  | 109        |
| <b>Summary</b>   | <b>110</b> |

|   |            |
|---|------------|
| <b>Chapter 6: Personalizing E-mail Body</b>                     | <b>111</b> |
| <b>Enhancing messages using built-in placeholders</b>           | <b>111</b> |
| Placeholders in confirmation messages                           | 113        |
| <b>Personalizing messages using member attributes</b>           | <b>113</b> |
| <b>Sending messages to subsets of lists based on attributes</b> | <b>115</b> |
| Increasing the amount of criteria available                     | 116        |
| <b>Using message templates</b>                                  | <b>117</b> |
| Creating a message template                                     | 117        |
| <b>Summary</b>  | <b>120</b> |
| <b>Chapter 7: Measuring Effectiveness of Newsletters</b>        | <b>121</b> |
| <b>User tracking</b>  | <b>122</b> |
| Sending a message with user tracking enabled                    | 122        |
| Examining user-tracking statistics per message                  | 123        |
| Examining user-tracking statistics per user                     | 124        |
| <b>Click tracking</b>   | <b>125</b> |
| Avoiding false spam / phishing detection                        | 126        |
| Enabling click tracking   | 128        |
| Enabling extra detail in click tracking                         | 128        |
| Sending an e-mail with click tracking enabled                   | 128        |
| Examining basic click-tracking statistics per message           | 129        |
| <b>Examining advanced user/click-tracking statistics</b>        | <b>130</b> |
| Extra statistics sidebar link                                   | 130        |
| Overview  | 131        |
| Viewing clicks by URL   | 132        |
| Viewing clicks by message                                       | 132        |
| Viewing opens by message  | 133        |
| Viewing domain statistics                                       | 133        |
| <b>Summary</b>  | <b>134</b> |
| <b>Chapter 8: Securing phpList</b>                              | <b>135</b> |
| <b>Changing the admin password</b>                              | <b>135</b> |
| Choosing strong passwords                                       | 135        |
| Changing phpList admin password                                 | 136        |
| <b>Confirming correct filesystem permissions</b>                | <b>137</b> |
| Confirming permissions in a GUI SFTP/FTP client                 | 137        |
| Confirming permissions using a shell session                    | 138        |
| Setting appropriate permissions                                 | 138        |
| <b>Confirming htaccess restrictions are in place</b>            | <b>139</b> |
| <b>Securing admin pages with additional (htauth) password</b>   | <b>139</b> |
| Creating an htpasswd file                                       | 140        |

---

|   |            |
|---|------------|
| Creating an htpasswd file online                                      | 140        |
| Creating an htpasswd file in a Unix shell                             | 141        |
| Amending admin/.htaccess  | 141        |
| <b>Summary</b>  | <b>143</b> |
| <b>Chapter 9: Advanced Features</b>                                   | <b>145</b> |
| <b>Advanced bounce handling</b>                                       | <b>145</b> |
| Bounce rules and regular expressions                                  | 146        |
| Creating a new bounce rule  | 146        |
| Creating a new rule based on an existing bounce                       | 147        |
| Auto-generating new bounce rules                                      | 149        |
| Checking current rules against bounces                                | 150        |
| <b>Domain-based e-mail throttling</b>                                 | <b>151</b> |
| <b>Adding attachments to messages</b>                                 | <b>151</b> |
| Adding multiple files simultaneously                                  | 153        |
| Attaching files stored on your web server                             | 155        |
| Securing your attachments stored on the web server                    | 158        |
| <b>Automatically repeat messages</b>                                  | <b>158</b> |
| Sending a repeating message   | 159        |
| Forcing a repeating message to repeat before embargo                  | 159        |
| <b>Auto-generating messages from RSS feeds</b>                        | <b>160</b> |
| Enabling RSS support  | 161        |
| Associating an RSS feed with a list                                   | 162        |
| "Getting" new RSS items   | 163        |
| User requirements to receive RSS messages                             | 163        |
| Sending a message including RSS                                       | 165        |
| Setting your RSS message's schedule                                   | 165        |
| <b>Setting up processing automation</b>                               | <b>166</b> |
| Requirements  | 167        |
| Customize the CLI "wrapper"   | 167        |
| <b>Summary</b>  | <b>169</b> |
| <b>Chapter 10: Hacking phpList</b>                                    | <b>171</b> |
| <b>Substituting user attributes in the subject line</b>               | <b>171</b> |
| Caveat #1 – no attribute substitution for a third party               | 173        |
| Caveat #2 – "Forward message" page displays an un-substituted subject | 174        |
| <b>Sending system messages as HTML instead of text</b>                | <b>174</b> |
| Code changes  | 174        |
| Creating the plain-text part of the message by stripping out the HTML | 175        |
| Sending both the HTML and text versions of the message                | 175        |
| Stripping slashes from messages stored in the database                | 176        |

|  |            |
|--|------------|
| Web interface changes  | 179        |
| Standard system messages   | 179        |
| Subscription and confirmation messages on the pre-existing subscribe page                | 181        |
| Tips   | 181        |
| <b>Creating a messages archive page</b>  | <b>183</b> |
| Customizing the number of messages displayed per page                                    | 184        |
| Adding a message summary to the list   | 185        |
| Hiding the summary in outgoing e-mails   | 186        |
| Linking the archives to the main index   | 186        |
| <b>Sending messages using your e-mail client</b>   | <b>188</b> |
| Installing MailToList  | 188        |
| Configuring the Mail To List   | 190        |
| Whitelisting users   | 192        |
| Manually adding users  | 192        |
| Importing users  | 193        |
| Processing incoming e-mails  | 193        |
| Automating Mail To List using cron   | 194        |
| Bugs in Mail To List   | 195        |
| Plain text messages will come out blank  | 195        |
| Plain text or HTML attachments will disappear  | 195        |
| <b>Summary</b>   | <b>196</b> |
| <b>Chapter 11: Troubleshooting and Maintenance</b>                                       | <b>197</b> |
| <b>General fault diagnosis</b>   | <b>197</b> |
| Test mode (the "dry run")  | 197        |
| Verbose mode – tell me what you're thinking  | 198        |
| <b>Common errors and warnings</b>  | <b>199</b> |
| Error: Please make sure that index.php is your default document for a directory          | 199        |
| Error: IMAP is not included in your PHP installation, cannot continue                    | 200        |
| Fatal error: Cannot connect to database, access denied. Please contact the administrator | 201        |
| HTTP Error 500: Internal server error  | 202        |
| HTTP Error 404: File not found   | 202        |
| Warning: The pageroot in your config does not match the current location                 | 203        |
| Warning: In safe mode, not everything will work as expected                              | 204        |
| Warning: The attachment repository does not exist or is not writable                     | 204        |
| Warning: open_basedir restrictions are in effect   | 205        |
| Warning: Things will work better when PHP magic_quotes_gpc = on                          | 205        |
| Warning: Things will work better when PHP magic_quotes_runtime = off                     | 206        |
| Warning: You are trying to use RSS, but XML is not included in your PHP                  | 206        |
| Mailer Error: SMTP Error: Could not connect to SMTP host                                 | 207        |

|  |            |
|--|------------|
| <b>Creating a backup</b>                       | <b>207</b> |
| Backing up the database                        | 207        |
| Backing up the files                           | 209        |
| <b>Upgrading phpList</b>                       | <b>209</b> |
| <b>Where to find more help and information</b> | <b>211</b> |
| <b>Summary</b>                                 | <b>212</b> |
| <b>Index</b>                                   | <b>213</b> |

---



# Preface

Tired of an e-mail BCC list that scrolls off the page or fiddly and hard-to-manage bulk mailing systems? You need phpList—a high-powered, robust, feature-packed mailing system that will "get out of your way" and get the job done.

*phpList 2 E-mail Campaign Manager* will guide you from basic installation and setup through management, reporting, and automation of phpList, the world's most popular open source e-mail campaign manager. It also covers advanced customization and configuration of phpList.

phpList is a popular open source e-mail campaign manager, sporting a powerful web frontend, a rich message editor, and an advanced feature set. We start with a basic configuration and finish with a full-featured e-mail management engine.

You will work your way up from the installation to advanced topics such as bounce automation, user and click-through tracking, and integration with third party publishing platforms such as WordPress, Drupal, Blogger, and Facebook.

Advanced topics such as securing your installation against spammers, attacks, and vulnerabilities are covered, as well as additional advanced and "experimental" features offered by phpList.

This book is an invaluable guide for an e-mail publisher who wants a robust and powerful engine to manage their small-to-huge e-mail distribution empire.

## What this book covers

*Chapter 1, Installation and Configuration*, discusses installation and basic configuration of phpList.

*Chapter 2, Setting up Subscriber Forms*, talks about creating forms to capture subscribers' details, preferences, and so on.



*Chapter 3, Setting up E-mail Bounce Handling*, includes automatically managing and reporting on permanent and temporarily bouncing e-mails.

*Chapter 4, Setting up phpList with Popular Publishing Platforms*, describes the integration of phpList with Drupal, WordPress, and Facebook.

*Chapter 5, List, User, and Administrator Management*, talks about managing users, admins, and lists and the Performing bulk actions (import/export).

*Chapter 6, Personalizing E-mail Body*, includes using attributes and variables to customize e-mails received by subscribers.

*Chapter 7, Measuring Effectiveness of Newsletters*, describes tracking open and click-through rates, and reporting on the overall effectiveness of each message.

*Chapter 8, Securing phpList*, discusses how to ensure you have basic protection from spammers, brute-force attacks, and misconfigurations.

*Chapter 9, Advanced Features*, talks about advanced bounce / message handling, RSS and file attachment, automatically repeating messages, and process automation.

*Chapter 10, Hacking phpList*, is about adding extra functionality and features by modifying or adding to phpList code.

*Chapter 11, Troubleshooting and Maintenance*, discusses common errors and maintenance tasks.

## What you need for this book

While phpList will technically run on a wide range of systems, this book has been written with the "ideal" environment in mind – a Linux hosting platform running Apache, PHP, and MySQL (this is the common environment offered by online shared or dedicated hosting providers).

You will need write access to such an environment, as well as the ability to transfer files to the hosting environment. Shell access to your hosting environment would also be useful, but not essential.

This book was written using phpList 2.10.12 as a basis for screenshots and tutorials. In future versions of phpList, user interface elements may change, including layout and text, and additional features may be introduced.

## Who this book is for

This book is aimed at content creators and distributors who want to "up their game" to a full e-mail distribution engine.

Further, more it will certainly appeal to you if you are technically inclined and would rather "roll your own" system than use an off-the-shelf product.

It is assumed that you are familiar with the basic concepts of e-mail, editing text files, and managing remote files with FTP / SFTP.

The more advanced topics will teach you how to get your "hands dirty" in some of phpList's open source code.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Open `admin/subscribelib2.php` and search for `spambot`."

A block of code is set as follows:

```
define('USE_DOMAIN_THROTTLE',0);
define('DOMAIN_BATCH_SIZE',1);
define('DOMAIN_BATCH_PERIOD',120);
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:


```
define('USE_DOMAIN_THROTTLE',0);
define('DOMAIN_BATCH_SIZE',1);
define('DOMAIN_BATCH_PERIOD',120);
```

Any command-line input or output is written as follows:

```
Process bounces: /path/to/bin/phplist -p processbounces
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Click on **add new** to start adding attribute values."

 Warnings or important notes appear in a box like this. ]

 Tips and tricks appear like this. ]

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on [www.packtpub.com](http://www.packtpub.com) or e-mail [suggest@packtpub.com](mailto:suggest@packtpub.com).

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code for this book

You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.



# 1

## Installation and Configuration

phpList is a popular open source e-mail campaign manager, sporting a powerful web frontend, rich message editor, and advanced feature set.

Throughout this book, you will be guided from the basics of initial configurations to making advanced customizations and modifications.

The first task required in building our phpList system is the initial installation and configuration of phpList on a web server.

Let's get started!

In this chapter, you will learn how to install and configure a basic phpList system, including:

- Installing phpList files and database
- Making configuration changes
- Basic configuration of lists, users, and attributes
- Sending your first message

### Installing phpList files and database

The phpList system contains HTML and image files, as well as the PHP code, which comprises the phpList. These are static files, and not expected to change, unless you upgrade, or otherwise manually modify them.

The database is where the dynamic content is stored, including lists, members, and messages.

We will install the phpList files and then use a setup process to have phpList set up the database.

## Requirements

For the purposes of this book, we will limit ourselves to the developer-tested system requirements detailed at [http://www.phplist.com/system\\_requirements](http://www.phplist.com/system_requirements), namely:

- A Linux-based platform
- The Apache web server
- MySQL database (version 4.0+)
- PHP (version 4.3+)

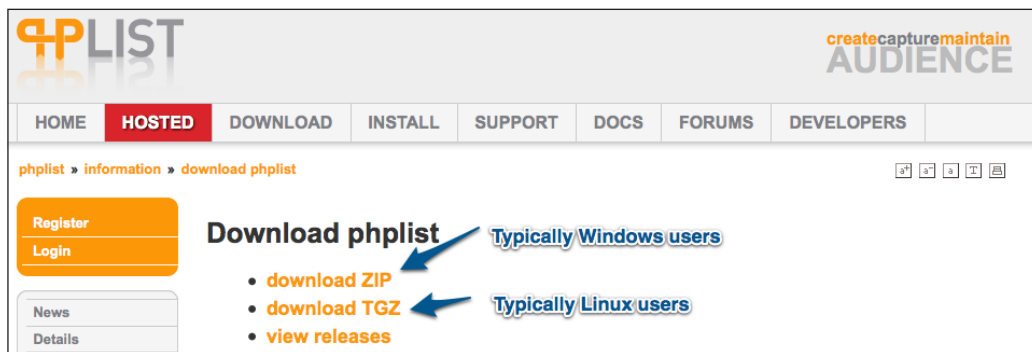
 As it will be discussed in detail in *Chapter 3, Setting up E-mail Bounce Handling*, phpList's bounce handling also requires the `php-imap` module.

While phpList has been reported to work on Windows servers, by using either IIS or an Apache/PHP/MySQL environment, this book and the examples herein will focus on the Linux-based system described previously.

## Downloading phpList

Our first port of call is the phpList download page (<http://www.phplist.com/download>). Here we can download the latest version of phpList, packaged either as a ZIP file (typically for Windows users) or TGZ file (typically for Linux users).

Clicking either of the links will auto-start a download of the latest version of phpList in your preferred packaging format:



## Unpacking and uploading phpList

Unpack the .ZIP or .TGZ file into a new folder. You'll end up with a directory structure looking something as follows:

| Name               | Date Modified       | Size   |
|--------------------|---------------------|--------|
| bin                | 3/02/2011 6:48 AM   | --     |
| scripts            | 3/02/2011 6:48 AM   | --     |
| public_html        | 3/02/2011 6:48 AM   | --     |
| VERSION            | 3/02/2011 6:38 AM   | 4 KB   |
| phplist.php        | 4/09/2010 7:10 AM   | 16 KB  |
| PEOPLE             | 21/04/2010 1:56 AM  | 8 KB   |
| INSTALL            | 28/08/2007 2:38 AM  | 4 KB   |
| readme_ja.txt      | 26/07/2007 9:46 AM  | 4 KB   |
| UPGRADE            | 13/09/2005 6:38 AM  | 4 KB   |
| README.usertables  | 13/09/2005 6:38 AM  | 4 KB   |
| README.rssfeeds    | 13/09/2005 6:38 AM  | 4 KB   |
| README.passwords   | 13/09/2005 6:38 AM  | 4 KB   |
| COPYING            | 13/09/2005 6:38 AM  | 20 KB  |
| phplist.sql        | 9/09/2005 8:07 AM   | 168 KB |
| README             | 30/08/2005 12:55 PM | 4 KB   |
| BUGS               | 3/08/2005 1:37 PM   | 4 KB   |
| README.batches     | 4/04/2004 10:19 AM  | 4 KB   |
| README.commandline | 7/03/2004 3:09 AM   | 4 KB   |
| README.security    | 5/03/2004 12:29 AM  | 4 KB   |
| README.repetition  | 26/02/2004 3:20 AM  | 4 KB   |
| TODO               | 25/11/2003 4:45 AM  | 4 KB   |
| README.bounces     | 25/11/2003 4:45 AM  | 4 KB   |
| NEWS               | 25/11/2003 4:45 AM  | 4 KB   |
| changelog.presf    | 22/11/2003 9:53 AM  | 33 KB  |

Note the three folders named `bin`, `scripts`, and `public_html`. Of these folders, only the `public_html` folder is required for phpList (like its name, these are the files which will be publicly available on your web server). The other folders are complimentary and contain useful scripts, documentation, and so on.



The structure of the `public_html` folder assumes that you're going to put phpList at the `/lists/` location of your URL—that is, your phpList installation will be accessed at `http://www.yourdomain.com/lists/`.

If you are satisfied with the default `/lists/` location, then simply upload the entire `lists` folder directly to your web server, or else upload the contents of the folder to the preferred location where you want the phpList to be installed on your web server. Remember that this location must be publicly accessible (that is, it must be under the same directory structure as the rest of your publicly accessible web content).



## Creating a database

How you create a database for phpList will depend on your web hosting provider, but after the database creation, the following information will be required to configure phpList:

- Database server hostname (often localhost)
- Database username
- Database password
- Database name

## Performing initial configuration

phpList's configuration file is `config/config.php`. This file is extensive, and contains many advanced settings, which will be dealt with later in this book. For starters, we'll configure some of the basic settings to get our phpList installation running.

### Database settings

Starting from line #22, customize the following four entries for the database you created before:

```
$database_host = "localhost";
```

Change this to the name of your database host. Leave it as "localhost" if you're not sure – if this doesn't work, contact your hosting provider.

```
$database_name = "phplistdb";
```

Change this to the name of the database you created.

```
$database_user = "phplist";
```

Change this to the user you created to access the database.

```
$database_password = "phplist";
```

Change this to the password of the user you created.

### Path settings

If you chose to install phpList to a path other than `/lists` when you uploaded it, you'll also need to change the following two entries, starting at line #48:

```
$pageroot = '/lists';
```

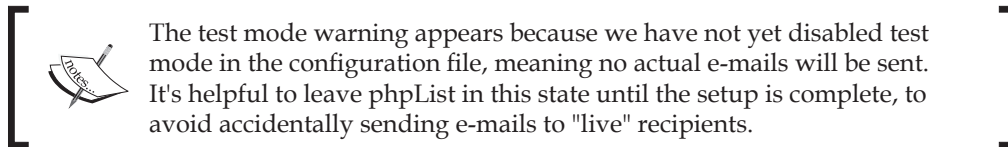
Change this to whatever you selected as your phpList path. This could simply be "/" or something else, like "/newsletters"

```
$adminpages = '/lists/admin';
```

Change this to match the \$pageroot aforementioned definition. phpList requires that the "admin" portion remains the same, so this must be whatever your \$pageroot is, with "/admin" added to the end, for example, "/newsletters/admin".

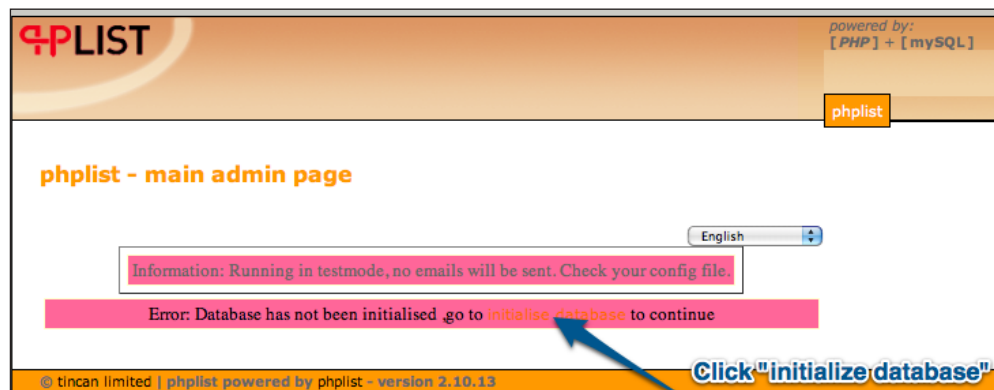
## Performing web-based configuration

Access your phpList admin section at the location where you installed it, appending "/admin" to the URL, as follows: `http://www.yourdomain.com/lists/admin`.

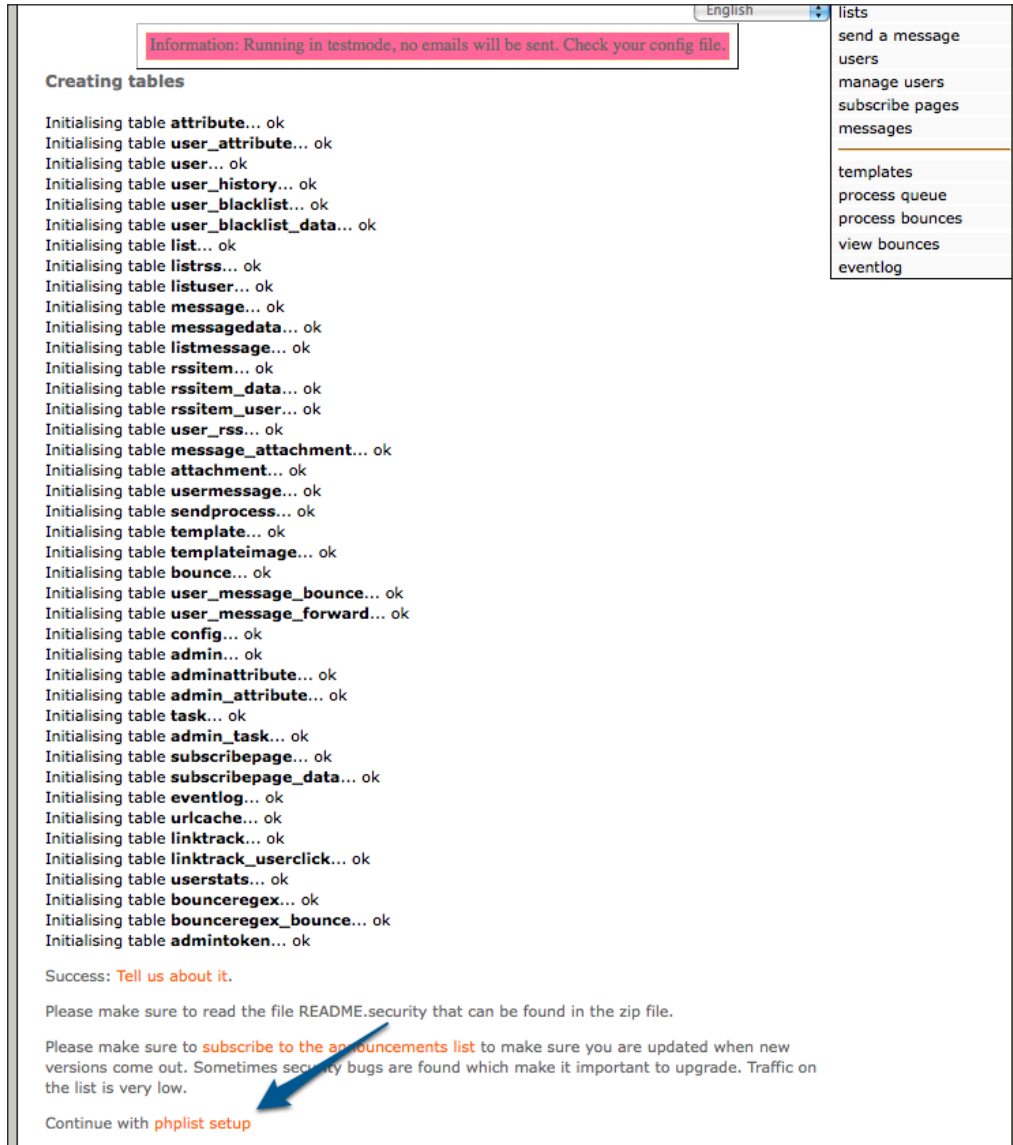


## Initialize database

Although it is hard to see because of the pink highlighting, you are warned that the database has not yet been initialized, so click on **initialize database** to continue:



The database initialization script will automatically connect to the database using the username and password you specified, and will create and populate all the required database tables. If you see any errors, check to see if your database settings are correct. A successful initialization will look like the one in the following screenshot. Having successfully initialized the database, click on the **phplist setup** link at the bottom of the page to continue:



## Initial login

You're now taken to the admin login screen and presented with a user/password request. The default username is **admin** and the password is **phplist**. Enter this combination to proceed:

powered by:  
[PHP] + [MySQL]

phplist

- main page
- about phplist

English

This document requires you to log in

Default login is admin, with password phplist

Name:  
admin

Password:  
\*\*\*\*\*

Enter

Forgot Password?: Enter your email:  Send Password

Default password is "phplist"

You are then presented with a quick checklist of steps to complete your phpList setup. Start by clicking on **go there** next to **Change Admin Password** to change your password:

powered by:  
[PHP] + [MySQL]

phplist

- logout
- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

logout

English

Information: Running in testmode, no emails will be sent. Check your config file.

This page show a quick checklist of things to set up in order to get phplist working correctly

- Initialise Database go there
- Change Admin Password go there X
- Configure General Values go there X
- Configure Attributes go there X
- Create Lists go there X
- Create Subscribe Pages go there X

Click to change password

## Change Admin Password

Next, you are taken to the manage admins page (we'll cover this in detail in *Chapter 5, List, User, and Administrator Management*) and prompted to change the password for the "admin" user.

1. It's a good idea to associate your e-mail address with this user at the same time. Update the **Password** and the **Email** address and click on **Save Changes**:

Set permissions to view pages in the system:

- All: admin has access to page without restrictions
- View: admin can view content of pages, but not change anything. This currently only works for the "user", "users" and "members" pages.
- None: admin cannot see this page
- Owner: admin can see this page, but only see the content of the lists they own

[list of administrators](#)

manage users  
subscribe pages  
messages  
templates  
process queue  
process bounces  
view bounces  
eventlog

Edit Administrator: admin

Admin Details:

|                                |                          |
|--------------------------------|--------------------------|
| ID                             | 1                        |
| Login Name                     | admin                    |
| Normalised loginname           | admin                    |
| Email                          | myemailaddress@myisp.com |
| Time Created                   | 2011-03-20 12:11:01      |
| Time modified                  | 2011-03-20 12:11:01      |
| Modified by                    |                          |
| Password                       | mysecretnewpassword      |
| Last time password was changed | 2011-03-20               |
| Is this user Super Admin?      | 1                        |
| Is this account disabled?      |                          |

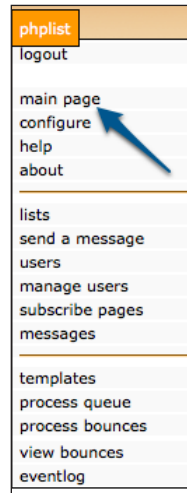
Save Changes

2. You'll notice that when you click on **Save Changes**, the current page is simply reloaded with a message confirming that the changes have been saved:

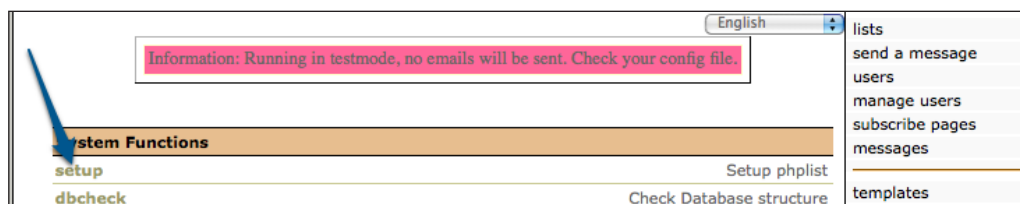
Information: Changes saved

Edit Administrator: admin

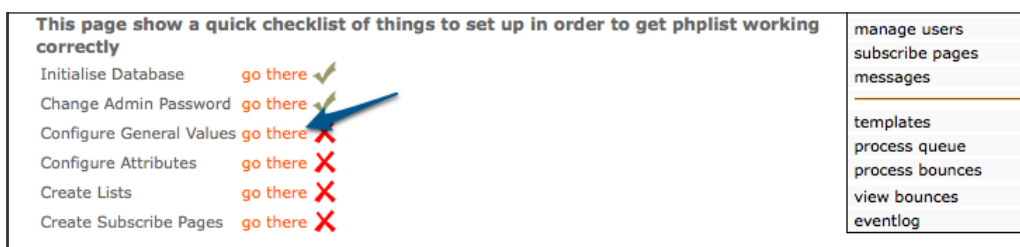
- To continue with the setup, click on the **main page** link on the right-hand navigation panel:



- We haven't completed the initial checklist yet, so click on the **setup** link to continue:



- The next item on our checklist is **Configure General Values**, so click on **go there** to continue:





## Configure General Values

You are now taken to the configuration page. This can be accessed in future from the **configure** link on the right-hand navigation panel. These are the configuration values that are stored in the database, as opposed to the `config/config.php` file.

There are a few general attributes we should check right away – the rest of them can be tweaked later:

- Website address (without `http://`) – This value would have been auto-detected based on the URL which you used to perform the database initialization. Ensure it is correct, because this value is used to create the internal hyperlinks on the web interface.

 If you ever move your phpList installation to another URL, remember to update this value. 

- Person in charge of this system (one e-mail address) – Make sure this address is valid. Notifications of new subscriptions and other system actions will be sent to this address.

You can change a number of other configuration settings, like additional e-mail addresses to receive system notifications, default size of WYSIWYG editor windows, default welcome/confirmation messages, and so on.

Click on the **main page** link in the right-hand panel to return to the main page. Then click on the **setup** link to continue the initial setup.

Our next step is the **Configure Attributes** item, so click on **go there** to continue:

|   |            |                 |
|---|------------|-----------------|
| <b>This page show a quick checklist of things to set up in order to get phplist working correctly</b> |            | manage users    |
| Initialise Database   | go there ✓ | subscribe pages |
| Change Admin Password   | go there ✓ | messages        |
| Configure General Values  | go there ✓ | templates       |
| Configure Attributes  | go there ✓ | process queue   |
| Create Lists  | go there ✗ | process bounces |
| Create Subscribe Pages  | go there ✗ | view bounces    |
|   |            | eventlog        |


## Configure Attributes

Attributes are used to record additional data about subscribers (such as, name, age, and so on), and to make selective decisions about e-mailing them based on these attributes (that is, you may send a different message to all female subscribers on Mothers' Day).

1. To get started, click on the **predefined defaults** link to import some pre-prepared attributes:

Here you can identify the attributes you want people to enter, in order to sign up for the lists. Attributes are "global" ie, they apply to all lists.

[Add a new one](#)

Load data from [predefined defaults](#) 

No Attributes have been defined yet

[Save Changes](#)

**Add new Attribute:**

|                              |                                       |
|------------------------------|---------------------------------------|
| Name:                        | <input type="text"/>                  |
| Type:                        | <input type="text" value="textline"/> |
| Default Value:               | <input type="text"/>                  |
| Order of Listing:            | <input type="text"/>                  |
| Is this attribute required?: | <input checked="" type="checkbox"/>   |

[Save Changes](#)

[manage users](#)  
[subscribe pages](#)  
[messages](#)  


---

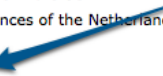
[templates](#)  
[process queue](#)  
[process bounces](#)  
[view bounces](#)  
[eventlog](#)

2. Choose the **Countries in the world** attributes and click on **Add**:

**Add some possible useful attributes**

Choose the attributes you want to add to your mailinglist system:

- Provinces of Belgium
- Estados en México
- Countries in the world**
- Counties in the UK
- Estados Brasileiros
- States in India
- Provinces in Canada
- Cities of Belgium
- States in the USA
- Provinces of the Netherlands
- Sexo

[Add](#) 

[manage users](#)  
[subscribe pages](#)  
[messages](#)  


---

[templates](#)  
[process queue](#)  
[process bounces](#)  
[view bounces](#)  
[eventlog](#)



3. You can see a simple confirmation message, noting that the attributes have been added. To re-examine these attributes, return to the main admin page by clicking on the **main page** link:

The screenshot shows the phplist admin interface. At the top, it says "phplist - some useful default attributes" and "good afternoon admin". There is a "logout" link. A language dropdown menu is set to "English". A pink information box states: "Information: Running in testmode, no emails will be sent. Check your config file." Below this, it says "Add some possible useful attributes" and "Choose the attributes you want to add to your mailinglist system:". A status message says "Loading Countries in the world Done". On the right, there is a navigation menu with links: "main page", "configure", "help", "about", "lists", "send a message", "users", "manage users", "subscribe pages", "messages", and "templates". A blue arrow points to the "main page" link.

4. Note that **Country** is now displayed under the **Configuration functions** section. Click on the **setup** link to continue the setup process:

The screenshot shows the "System Functions" and "Configuration functions" sections. Under "System Functions", there are links for "setup" (Setup phplist), "dbcheck" (Check Database structure), and "eventlog" (View the eventlog). Under "Configuration functions", there are links for "configure" (configure phplist), "attributes" (Configure Attributes), "Country" (Control values for Country), and "spage" (Configure Subscribe pages). A blue box highlights the "Country" link, and a blue arrow points to it. A tooltip above the "Country" link says "Country now appears". On the right, there is a navigation menu with links: "subscribe pages", "messages", "templates", "process queue", "process bounces", "view bounces", and "eventlog".

5. The next item on our checklist is **Create Lists**, so click on **go there** to proceed:

The screenshot shows a checklist of tasks to set up phplist correctly. The tasks are: "Initialise Database" (go there ✓), "Change Admin Password" (go there ✓), "Configure General Values" (go there ✓), "Configure Attributes" (go there ✓), "Create Lists" (go there ✗), and "Create Subscribe Pages" (go there ✗). A blue arrow points to the "go there ✗" link for "Create Lists". On the right, there is a navigation menu with links: "manage users", "subscribe pages", "messages", "templates", "process queue", "process bounces", "view bounces", and "eventlog".

## Create Lists

You're now taken to the **lists** page. This page can also be accessed using the **lists** link on the right-hand navigation panel.

Currently, there is only a single default list, named **test**. This list is inactive. We'll use this to test our phpList installation, so click the **Active** checkbox to make it active, and click on **Save Changes** to apply:

The screenshot shows a table with columns: No, Name, Order, Functions, Active, and Owner. The first row contains the list 'test' with order '0', functions 'edit | view members | delete', an unchecked 'Active' checkbox, and owner 'admin'. Below the table is a 'Save Changes' button and a link 'add a list'. On the right, a navigation menu includes 'manage users', 'subscribe pages', 'messages', 'templates', 'process queue', 'process bounces', 'view bounces', and 'eventlog'. Blue arrows point to the 'Active' checkbox and the 'Save Changes' button.

| No | Name | Order | Functions                    | Active                   | Owner |
|----|------|-------|------------------------------|--------------------------|-------|
| 1  | test | 0     | edit   view members   delete | <input type="checkbox"/> | admin |

0 Members  
List for testing.

[Save Changes](#)

[add a list](#)

- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

The lists page will be refreshed with your changes applied. Click on the **main page** link to return to the main page, and then click once more on the **setup** link to complete the last item on the checklist.

The final item is labeled **Create Subscribe Pages**, so click on **go there** to proceed:

The screenshot shows a checklist titled 'This page show a quick checklist of things to set up in order to get phplist working correctly'. The items are: 'Initialise Database' (go there ✓), 'Change Admin Password' (go there ✓), 'Configure General Values' (go there ✓), 'Configure Attributes' (go there ✓), 'Create Lists' (go there ✓), and 'Create Subscribe Pages' (go there ✗). A blue arrow points to the 'go there ✗' link for 'Create Subscribe Pages'. On the right, a navigation menu is identical to the previous screenshot.

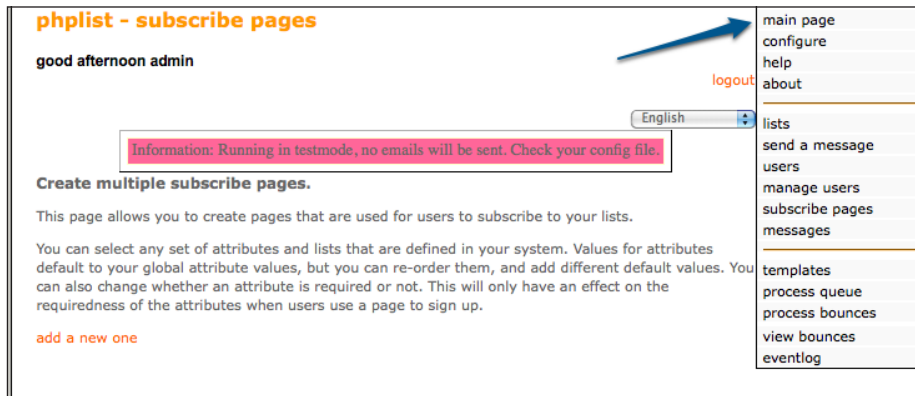
**This page show a quick checklist of things to set up in order to get phplist working correctly**

- Initialise Database [go there](#) ✓
- Change Admin Password [go there](#) ✓
- Configure General Values [go there](#) ✓
- Configure Attributes [go there](#) ✓
- Create Lists [go there](#) ✓
- Create Subscribe Pages [go there](#) ✗

- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

## Create Subscribe Pages

You are now taken to the **subscribe pages** page, which can also be reached using the navigation panel. We will cover **subscribe pages** in detail in the next chapter, so let's ignore this for now and return to the main page to test message delivery!



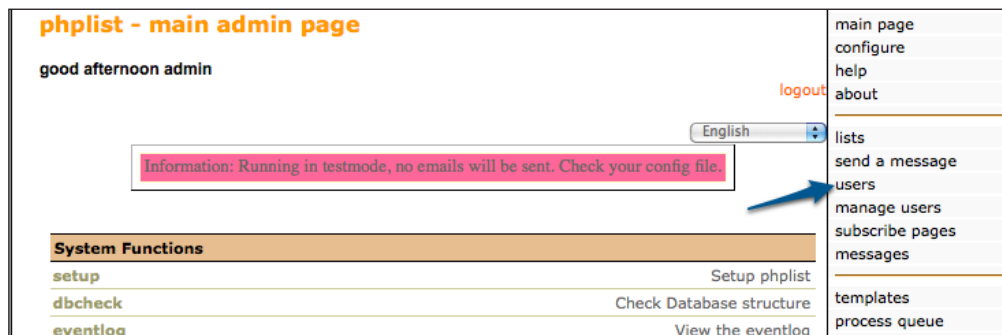
## Testing e-mail delivery

While we'll cover subscribe pages, bounce handling, and e-mail personalization in subsequent chapters, let's send ourselves a test message to ensure that we're complete with our basic setup.

## Create user

As we don't have subscribe pages for automatic user creation (and as we're the administrator anyway), let's add a sample user:

1. Click on the **users** link in the navigation panel to go to the users administration page:



2. Here we see that we currently have zero users. Click on **add a user**:

0 users in total  
Users marked red are unconfirmed (0)

Show only unconfirmed users:   
Show only blacklisted users:

Sort by:  desc:  asc:

Find a user:

[download all users as csv file](#)

[add a user](#)

- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

3. To add a new user, complete at least the following fields:
- **Email** (the user's e-mail address)
  - **Is this user confirmed** (set to **1** to create this user as confirmed, opted-in)
  - **Mailinglist Membership** (check the **test** list so that we can send e-mails to this user)
4. You can also complete the following optional fields:
- **Send this user HTML emails** (set to **1** to enable HTML for this user)
  - **Country** (one of the country attributes we imported earlier)
5. Click on **Save Changes** to apply.

[back to the list of users](#)  
Add a new User  
User Details

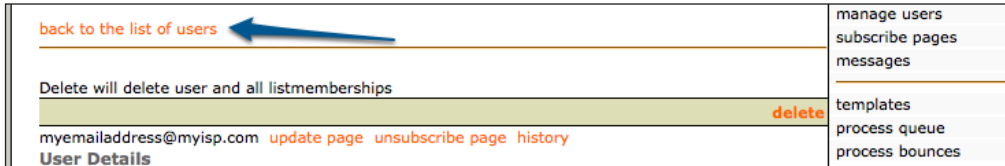
|                                  |                          |
|----------------------------------|--------------------------|
| ID                               |                          |
| Email                            | myemailaddress@myisp.com |
| Is this user confirmed (1/0)     | 1                        |
| Is this user blacklisted         | 0                        |
| Number of bounces on this user   |                          |
| Entered                          |                          |
| Last Modified                    |                          |
| Unique ID for User               |                          |
| Send this user HTML emails       | 1                        |
| Which page was used to subscribe |                          |
| RSS Frequency                    |                          |
| Password                         |                          |
| Last time password was changed   |                          |
| Is this account disabled?        |                          |
| Additional data                  |                          |
| Foreign Key                      |                          |
| Country                          | Antarctica               |

Mailinglist Membership:

test

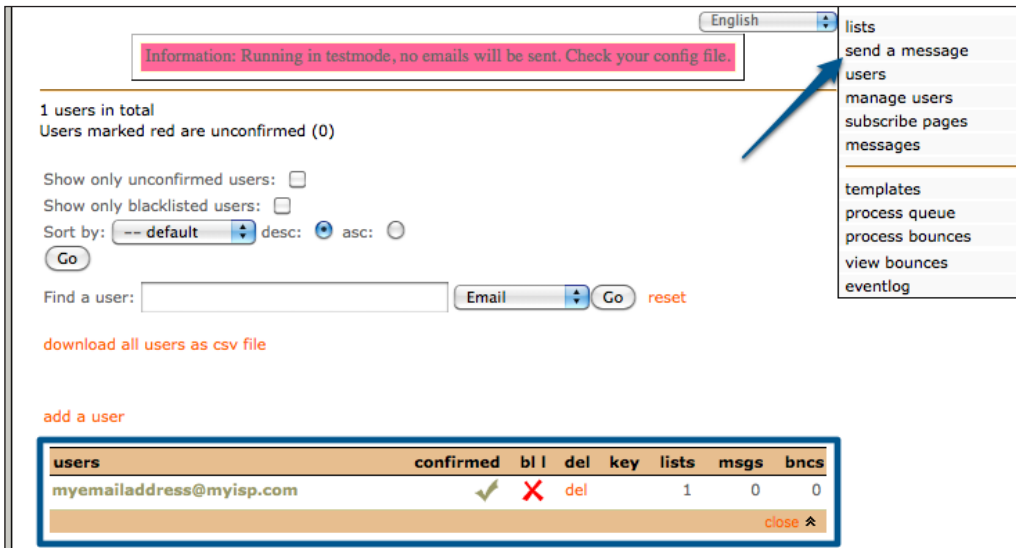
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

On saving, the page will refresh, with the user's details permanently on the page, as well as some additional administration links (unsubscribe, history, and so on). Click on **back to the list of users** to return to the user list:



The user list now shows that we have a single confirmed user, who's subscribed to one list.

Click on the **send a message** link in the right-hand navigation panel to send your first message:



## Sending a message

To send the most basic of messages:

1. Enter a subject, and some content in the WYSIWYG editor:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

|                 |
|-----------------|
| subscribe pages |
| messages        |
| templates       |
| process queue   |
| process bounces |
| view bounces    |
| eventlog        |

**Content** Format Scheduling Criteria Lists Misc

? Subject:

? From Line:

? Message.

Source

**B** *I* U ABC x<sub>2</sub> x<sup>2</sup>

Style  Format  Font  Size

Hello world :-)

- Next, click on **Save Changes** at the very bottom of the page.
- The page will be refreshed, with your message saved in a draft state. Now click the **Lists** tab to choose which lists this message will be sent to:



- Choose the **test** list and then click on **Send Message to the Selected Mailinglists** to add this message to the queue:

**Content** Format Scheduling Criteria Lists Misc eventlog

Send Test Message to email address(es):   
(comma separated addresses - all must be users)

Save Changes

Please select the lists you want to send it to:

- All Lists
- All Active Lists
- test (List is Active)  
List for testing.

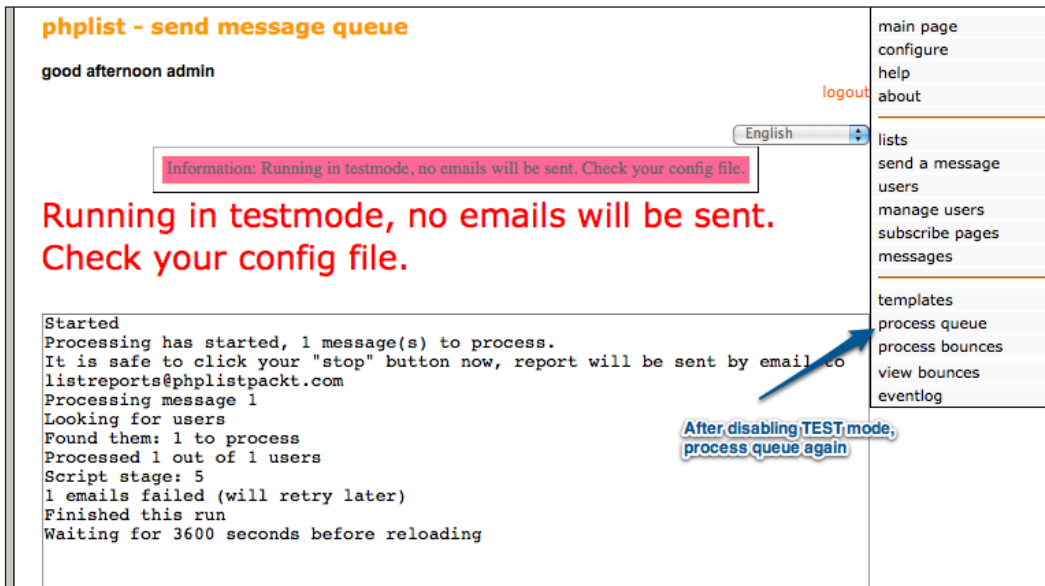
Send Message to the Selected Mailinglists

5. phpList will confirm that the message has been queued and offers you the option to manually process the queue. Click on the **process the message queue** link to process the queue:



## Processing the message queue (in test mode)

Because we haven't yet disabled test mode, no actual e-mail will be sent. phpList will make us aware of this in big, bold, red text. However, this does afford us the opportunity to check that everything else is working as expected:



## Disabling test mode

Now that we're confident that we've completed the initial setup correctly, let's disable the test mode, so that we can do a real e-mail delivery.

To disable the test mode, edit `config/config.php` on line #191 and change this line:

```
define ("TEST",1);
```

To:

```
define ("TEST",0);
```

Save the file and click on **process queue** again to send the e-mail for real.

## Processing the message queue (for real)

Once you have clicked on **process queue** again, your message will be processed for real. The test subscriber will receive their message and the list admin will receive a delivery report:

**phplist - send message queue**

good afternoon admin

logout

English

```
Started
Processing has started, 1 message(s) to process.
It is safe to click your "stop" button now, report will be sent by email to
listreports@phplistpackt.com
Processing message 1
Looking for users
Found them: 1 to process
Processed 1 out of 1 users
It took 8 mins 33 secs to send this message
Script stage: 5
1 messages sent in 0.79 seconds (4569 msgs/hr)
Finished this run
Waiting for 3600 seconds before reloading
```

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

## Summary

In this chapter, you've learned about how to install and configure phpList, as well as some basic administration tasks, including:

- Installing phpList files and database
- Making configuration changes
- Basic configuration of lists, users, and attributes
- Sending your first message

In the next chapter, we'll talk about enabling users to subscribe to your lists using subscribe pages and attributes.





# 2

## Setting up Subscriber Forms

Now that we have phpList installed and set up, it's time to set up subscriber forms so that our audience can sign up to our various newsletters. Using subscriber forms, you can not only encourage members to subscribe to your lists, but you can also capture valuable demographic information (attributes), and have your members set their own e-mail delivery preferences (HTML or text formatting, for example).

In this chapter, you will learn how to implement and customize subscriber forms. You will cover the following:

- Setting up a basic subscriber form
- Modifying a subscriber form
- Adding reCAPTCHA support to a subscriber form, by modifying phpList files
- Implementing a subscriber form on other applications (Blogger, WordPress, and so on)

### Basic concepts

As phpList supports multiple lists and multiple attributes, it's also possible to present a user with multiple subscribe pages. However, phpList must always have one default subscribe page. This is the page you'll see on a default installation if you browse to the default subscribe action, that is, <http://your-phplist-site/?p=subscribe>.

The following examples are based on these assumptions:

- You have created a list called "customers" with a description of "DVD Rental customers"
- Your phpList installation is at <http://your-phplist-site>

## Setting up a basic subscribe page

To view available subscriber pages, click on **subscribe pages** in the right-hand panel on the admin interface. On a default phpList installation, there are three pre-created subscribe pages, named **Email Only**, **Everything**, and **Name Only**.

1. We will start by using the **Email Only** subscribe page, so click on the radio button in the **default** column to make this subscribe page the default page:

**Create multiple subscribe pages.**

This page allows you to create pages that are used for users to subscribe to your lists.

You can select any set of attributes and lists that are defined in your system. Values for attributes default to your global attribute values, but you can re-order them, and add different default values. You can also change whether an attribute is required or not. This will only have an effect on the requiredness of the attributes when users use a page to sign up.

| Subscribe Pages | title      | edit | del | view | status | owner | default                          |
|-----------------|------------|------|-----|------|--------|-------|----------------------------------|
| 3               | Email Only | edit | del | view | active | admin | <input checked="" type="radio"/> |
| 2               | Everything | edit | del | view | active | admin | <input type="radio"/>            |
| 1               | Name Only  | edit | del | view | active | admin | <input type="radio"/>            |

add a new one

2. Now click on the **edit** link to confirm the lists associated with this subscribe page:

| Subscribe Pages | title      | edit | del | view | status | owner | default                          |
|-----------------|------------|------|-----|------|--------|-------|----------------------------------|
| 3               | Email Only | edit | del | view | active | admin | <input checked="" type="radio"/> |

3. Ignoring the customization options for now, scroll to the bottom of the page, ensure that our "customers" list is checked and that the rest of the default lists are unchecked, and click on **Save Changes** to apply the changes:

**Select the lists to offer**

test List for testing. I you don't make this list live and add yourself as a member, you can use this list to test a message. If the message comes out ok, you can resend it to other lists.

list 2

list 3

list 4

list 5

list 6

customers DVD Rental customers

Owner: admin

Save Changes Save and Activate Save and Deactivate



By default, if there is more than one list associated with a subscribe page, the user will be presented with a selection of lists to subscribe to. If there is only a single list associated with a subscribe page, the user will not see the name of the list they are subscribing to. This behavior is controlled by the **Display subscribe form automatically if only one active list is offered** option in the configuration page.

- Return to the **subscribe pages** list and click on **view** next to the **Email Only** list, to confirm how your new subscriber page will look:

You can select any set of attributes and lists that are defined in your system. Values for attributes default to your global attribute values, but you can re-order them, and add different default values. You can also change whether an attribute is required or not. This will only have an effect on the requiredness of the attributes when users use a page to sign up.

| Subscribe Pages | title      | edit | del | view | status | owner | default                          |
|-----------------|------------|------|-----|------|--------|-------|----------------------------------|
| 3               | Email Only | edit | del | view | active | admin | <input checked="" type="radio"/> |
| 2               | Everything | edit | del | view | active | admin | <input type="radio"/>            |
| 1               | Name Only  | edit | del | view | active | admin | <input type="radio"/>            |

subscribe pages  
 messages  


---

 templates  
 process queue  
 process bounces  
 view bounces  
 eventlog

- Because we are logged in as an administrator, we will see additional information on the subscriber page that regular users wouldn't see, and are able to bypass the "double-opt-in" feature and subscribe an e-mail address to our list immediately, without e-mail confirmation:

Subscribe to one or more of our mailing lists using the form below  
 required field

**You are logged in as administrator (admin) of this phplist system**

You are therefore offered the following choice, which your users will not see when they load this page.

[Go back to admin area](#)

**Please choose:**

Make this user confirmed immediately

Send this user a request for confirmation email

Email

Confirm your email address

6. If we direct our customers to `http://your-phplist-site/?p=subscribe`, they will be presented with the non-administrator version of this subscribe page and will be able to subscribe to our "customers" list.

We will create our own new subscriber form by clicking on the **add a new one** link at the bottom of the list of **subscribe pages**.

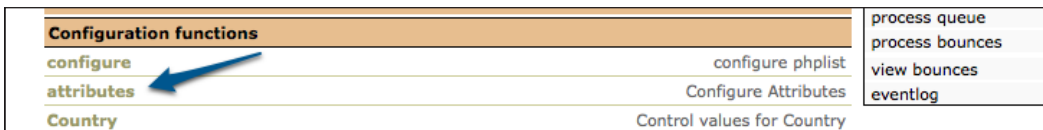
## Creating a custom subscribe page

Now that we know how to create a basic subscriber page, we may want to capture more information about our customers when they sign up. In this example, let's assume we'd like to know their name and movie genre preferences.

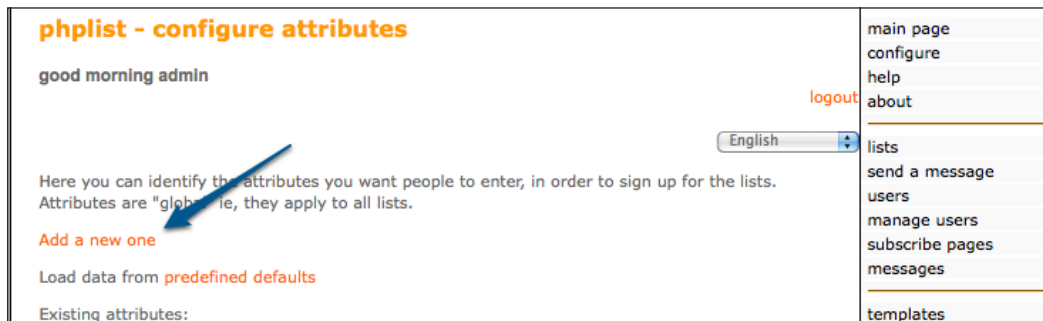
## Creating attributes

While we can take advantage of the built-in attributes, we can also create our own attributes to further personalize our e-mails. Let's create a new, optional attribute for the movie genre preference:

1. On the main admin page, click on the **attributes** link:



2. Click on **Add a new one** to create a new attribute:



- You are hyperlinked to the bottom of the page for the new attribute form. Enter a name for the attribute (**Movie Genre**) and an attribute type. In this case, **checkboxgroup** was chosen, because our readers may have more than one genre preference. Unless you want the attribute to be mandatory, remove the check from the checkbox next to **Is this attribute required?**. Click on **Save Changes** to create the new attribute:

**Add new Attribute:**

|                   |               |                              |                          |
|-------------------|---------------|------------------------------|--------------------------|
| Name:             | Movie Genre   |                              |                          |
| Type:             | checkboxgroup |                              |                          |
| Default Value:    |               |                              |                          |
| Order of Listing: |               | Is this attribute required?: | <input type="checkbox"/> |

[Save Changes](#)

- Once you've saved the changes, your attribute is created, but doesn't yet have any values for selection. To add values for selection, return to the main admin page, and click on the name of your attribute (**Movie Genre**) under the **Configuration functions** section:

| Configuration functions |                                  |
|-------------------------|----------------------------------|
| configure               | configure phplist                |
| attributes              | Configure Attributes             |
| Country                 | Control values for Country       |
| Favorite Food           | Control values for Favorite Food |
| <b>Movie Genre</b>      | Control values for Movie Genre   |
| spage                   | Configure Subscribe pages        |

- Click **add new** to start adding attribute values:

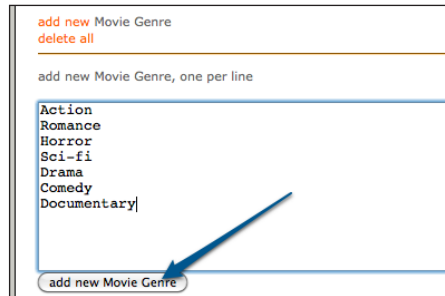
add new Movie Genre  
delete all

English

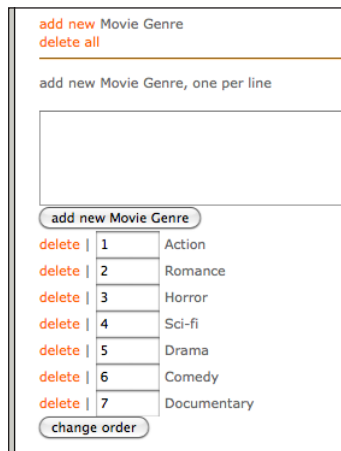
---

lists  
send a message  
users  
manage users  
subscribe pages

6. Add the attribute values, one per line, and then click on **add new Movie Genre** to save:



7. After saving your values, you can reorder, delete, or add new values from the same page:



## Creating subscribe page

Let's create a new subscribe page by clicking on **subscribe pages** from the right-hand navigation bar and then on the **add a new one** link at the bottom of the list of subscribe pages:



| Subscribe Pages | title      | edit                 | del                 | view                 | status | owner | default                          |
|-----------------|------------|----------------------|---------------------|----------------------|--------|-------|----------------------------------|
| 3               | Email Only | <a href="#">edit</a> | <a href="#">del</a> | <a href="#">view</a> | active | admin | <input checked="" type="radio"/> |
| 2               | Everything | <a href="#">edit</a> | <a href="#">del</a> | <a href="#">view</a> | active | admin | <input type="radio"/>            |
| 1               | Name Only  | <a href="#">edit</a> | <a href="#">del</a> | <a href="#">view</a> | active | admin | <input type="radio"/>            |

[add a new one](#)

## Customizing title and text

Name the new page **Movie Buffs** and enter custom intro text:

| General Information  |  |
|----------------------|--|
| Title                | <input type="text" value="Movie Buffs"/>   |
| Language file to use | --default  |
| Intro                | <pre>Welcome Movie Buffs, one and all! Enter your name, country, and genre preferences below, and we'll fill your inbox with exciting movie information!</pre> |

We can also customize the thank you page and the welcome/confirmation messages, but we'll leave them to the phpList default in this example.

## Customizing the HTML header and footer

Make a few minor adjustments to the HTML for the header and footer. In the following example, we will change the table width from 710 to 410 in the header, and add a custom H3 section "**Your number #1 source for movie rentals!**" to the footer:

|        |  |
|--------|--|
| Header | <pre>&lt;/head&gt; &lt;body bgcolor="#ffffff" background="images/bg.png"&gt; &lt;a name="top"&gt;&lt;/a&gt; &lt;div align=center&gt; &lt;table cellspacing=0 cellpadding=0 width=410 border=0&gt; &lt;tr&gt; &lt;td bgcolor="#000000" rowspan=3&gt;&lt;img height=1 alt="" src="images/transparent.png" width=1 border=0&gt;&lt;/td&gt; &lt;td bgcolor="#000000"&gt;&lt;img height=1 alt="" src="images/transparent.png" width=708 border=0&gt;&lt;/td&gt; &lt;td bgcolor="#000000" rowspan=3&gt;&lt;img height=1 alt=""</pre> |
| Footer | <pre>&lt;tr&gt; &lt;td colspan=3&gt; &amp;nbsp; &lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt;  &lt;/div&gt;&lt;h3&gt;Your number #1 source for movie rentals!&lt;/h3&gt; &lt;/body&gt;&lt;/html&gt;</pre>  |



These defaults are assigned to a new subscribe page when it is created. To change the defaults for any future subscribe pages, edit them under the **configure** page.



## Customizing delivery options

As we intend to deliver bright and colorful e-mails to our customers, let's set the default e-mail format to HTML, but allow the subscriber to opt for plain text e-mails with a checkbox:

|                   |                                  |  |
|-------------------|----------------------------------|--|
| HTML Email choice | <input type="radio"/>            | Don't offer choice, default to <b>text</b> |
|                   | <input type="radio"/>            | Don't offer choice, default to <b>HTML</b> |
|                   | <input checked="" type="radio"/> | Offer checkbox for text                    |
|                   | <input type="radio"/>            | Offer checkbox for HTML                    |
|                   | <input type="radio"/>            | Radio buttons, default to text             |
|                   | <input type="radio"/>            | Radio buttons, default to HTML             |

## Selecting attributes

Under the attribute selection section, select the **Name**, **Country**, and **Movie Genre** attributes:

| Select the attributes to use                     |  |
|--|--|
| Attribute:11                                     | Check this box to use this attribute in the page <input checked="" type="checkbox"/> |
| Name:  | <b>Name</b>  |
| Type:  | <b>textline</b>  |
| Default Value:                                   |  |
| Order of Listing: <input type="text" value="1"/> | Is this attribute required?: <input checked="" type="checkbox"/>                     |
| <hr/>  |  |
| Attribute:1                                      | Check this box to use this attribute in the page <input checked="" type="checkbox"/> |
| Name:  | <b>Country</b>   |
| Type:  | <b>select</b>  |
| Default Value:                                   |  |
| Order of Listing: <input type="text" value="2"/> | Is this attribute required?: <input type="checkbox"/>                                |
| <hr/>  |  |
| Attribute:10                                     | Check this box to use this attribute in the page <input checked="" type="checkbox"/> |
| Name:  | <b>Movie Genre</b>   |
| Type:  | <b>checkboxgroup</b>   |
| Default Value:                                   |  |
| Order of Listing: <input type="text" value="3"/> | Is this attribute required?: <input type="checkbox"/>                                |

## Selecting list

Finally, select the list to which this subscribe page will apply and **Save and Activate** your new list!

**Select the lists to offer**

test      List for testing. If you don't make this list live and add yourself as a member, you can use this list to test a message. If the message comes out ok, you can resend it to other lists.

list 2

list 3

list 4

list 5

list 6

DVD Rental customers

Owner:

View your new custom subscribe page to confirm that it looks as you want it to:

PHPlist

Welcome Movie Buffs, one and all! Enter your name, country, and genre preferences below, and we'll fill your inbox with exciting movie information!

required field

**You are logged in as administrator (admin) of this phplist system**

You are therefore offered the following choice, which your users will not see when they load this page.

[Go back to admin area](#)

**Please choose:**

Make this user confirmed immediately

Send this user a request for confirmation email

Email

Confirm your email address

I prefer to receive emails in Text format

Name

Country

Movie Genre

Action

Romance

Horror

Sci-fi

Drama

Comedy

Documentary

## Protecting your subscribe page from spammers

A major problem with online forms is the propensity of automated spam-bots to target them and fill your database or inbox with annoying false subscriptions and messages.

phpList includes a basic built-in level of spam protection, which is enabled by default. (See the `USE_SPAM_BLOCK` and `NOTIFY_SPAM` values in `config/config.php`).

To add an additional layer of spam-protection, we can incorporate reCAPTCHA (<http://www.google.com/recaptcha>), a free service that both serves to protect online forms from spammers and to help digitize books, newspapers, and so on.



Note that the following example requires changes to the phpList source code, which will be lost when performing an upgrade and the changes will, therefore, need to be reapplied after each successive upgrade.

## Signing up for reCAPTCHA keys and downloading the PHP library

The first step in implementing reCAPTCHA is to sign up (for free) to obtain a public / private key combination specific to your domain. Go to <http://www.google.com/recaptcha> and sign up.

Once you've been given your public and private key, record these somewhere safe and proceed to download the PHP library from <http://code.google.com/apis/recaptcha/docs/php.html>. As mentioned on the documentation page, you only need the `recaptchalib.php` from the library – the rest of the files are supplemental.

Extract `recaptchalib.php` from the downloaded zip file and save it to the `admin/` folder of your phpList installation. Be sure to confirm that your web server user has read-access to this file.

## Modifying `index.php`

Edit `index.php` (in the root of your phpList installation) and search for `USE_SPAM_BLOCK`. There is a line that reads:

```
if (USE_SPAM_BLOCK)
```

Insert the following code above this line:

```
// Insert reCAPTCHA if library exists and webserver has permission to
read it
if (file_exists('admin/recaptchalib.php') && is_readable('admin/
recaptchalib.php')) {
    require_once('admin/recaptchalib.php');
    $publickey = "your_public_key "; // you got this from the signup
page
    $html .= recaptcha_get_html($publickey);
}
```

## Modifying admin/subscribelib2.php

Edit admin/subscribelib2.php and search for spambot. There is a line that reads:

```
// anti spambot check
```

Insert the following code above this line:

```
// Confirm reCAPTCHA was successfully entered
if (file_exists('admin/recaptchalib.php') && is_readable('admin/
recaptchalib.php')) {
    require_once('admin/recaptchalib.php');
    $privatekey = "your_private_key ";
    $resp = recaptcha_check_answer ($privatekey,
        $_SERVER["REMOTE_ADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);
    if (!$resp->is_valid) {
        $missing = 'Please try typing the two words again';
        $allthere = 0;
    }
}
```

### Downloading the example code for this book



You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.


This will generate a reCAPTCHA input box on your subscribe form, and will return an error if a user tries to subscribe without correctly filling out the reCAPTCHA:



The screenshot shows a web form titled "PHPlist" with a header bar. Below the header, there is a message: "Subscribe to one or more of our mailing lists using the form below". A red error message reads: "required field The following required values are missing: Please try typing the two words again". The form contains two input fields: "Email" and "Confirm your email address", both containing the text "test@test.com". Below these fields is a reCAPTCHA widget showing the words "rusaff" and "century." in a stylized font. The reCAPTCHA interface includes a "Type the two words:" label, a text input field, and a "RECAPTCHA" logo with the tagline "stop spam. read books.". At the bottom of the form is a button labeled "Subscribe to the Selected Mailinglists".

## Putting a subscribe form on other applications

You may not always want to direct your customers to the phpList subscribe page, but may want (for example) to embed the subscribe form within an existing page on your blog.

[  This is covered in more detail in the phpList wiki at <http://docs.phplist.com/CustomSubscribeForm>. ]

## Generating HTML code for the subscribe form

Create your subscribe form within phpList, customizing it to suit your needs. Once you're satisfied with the way it looks, view the subscribe form as a normal user (that is, not as a logged-in administrator), and use your browser to view the source code. (Generally, right-clicking brings up a variant of a **view source** option in your browser).

Look in the source code for the following text:

```
<form method=post name="subscribeform">
```

This is the beginning of the HTML code for the subscribe form. Copy everything from the first "`<form`" to the ending "`</form>`" tag and paste it into a text editor.

## Customizing HTML code

Now we will customize this code to make it work as a generic form, which can then be embedded into an unrelated HTML page.

## Changing the form action

Change the following opening form tag from this:

```
<form method=post name="subscribeform">
```

To this, if this is the default subscribe page:

```
<form method=post action="http://your-phplist-site/?p=subscribe"
name="subscribeform">
```

Or to this, if this is not the default subscribe page, where `x` is the ID of the subscribe page:

```
<form method=post action="http://your-phplist-site/
?p=subscribe&id=x" name="subscribeform">
```

## Removing the JavaScript

Unless you manually copy the phpList JavaScript functions across to the website where you'll be embedding the form, you will probably want to remove anything in the form code between the `<script>` tags since, at best, these will be rendered ineffective and, at worst, will display errors.

## Optional pop-up confirmation

If you've embedded the subscribe form into another website, you may not want your reader to be redirected off your website and to the phpList "thank you" page. If we add a "target" to our form tag, our browser will bring up the "thank you" page in a new, pop-up window, leaving the original browser window unchanged.

Edit the HTML code and change the opening form tag from this:

```
<form method=post action="http://your-phplist-site/?p=subscribe"
name="subscribeform">
```

To this:

```
<form method="post" action="http://your-phplist-site/?p=subscribe"
name="subscribeform" target="windowName" onsubmit="window.open('',
this.target,
'dialog,modal,scrollbars=no,resizable=no,width=550,height=300,left=0,
top=0');">
```

## Removing the subscribe form button

You may want to further alter the form code in your HTML editor and perhaps replace the subscribe form button with an image. The form won't work without the original button though, so you can hide it by replacing:

```
<input type=submit name="subscribe" value="Subscribe to the Selected
Mailinglists" onClick="return checkform();">
```

With:

```
<input type="hidden" name="subscribe" value="yes">
```



### **Embedding code into other applications**

Having customized the HTML form code, you can now insert it into any HTML page to provide a subscription form.



## Summary

In this chapter, you've learned how to create and customize subscriber forms.

You've covered:

- Setting up a basic subscriber form
- Modifying the subscriber form
- Adding reCAPTCHA support to a subscriber form by modifying phpList files
- Implementing subscriber forms on other applications (Blogger, WordPress, and so on)

You'll notice, we've also covered three different ways of customizing phpList. The first is through the standard phpList admin interface, the second is through editing source code, and the third is through duplicating and customizing phpList-generated HTML code.

In the next chapter, we'll look at phpList automated bounce-handling features and at how to keep your lists clear of "stale" addresses.





# 3

## Setting up E-mail Bounce Handling

Now that we have forms enabling our subscribers to sign up, we need to keep our lists clean by implementing bounce handling. A **bounce** is an e-mail message which is returned because it is unable to be delivered.

In this chapter, you will learn how to configure and manage e-mail bounce handling including:

- Basic bounce settings
- Manually processing bounces
- Reviewing and acting on bounces

### E-mail bounce handling

phpList includes a robust bounce-handling system. It uses the "return-path" header of an e-mail to tell automated systems (that is, our subscribers' mail servers) who to inform about the failure to deliver our messages. This can be different from the "From" header in the e-mail, which is what the user sees when he/she reads our e-mail, and where out-of-office responses are sent.

### Requirements

phpList's bounce handling code requires the PHP "IMAP" module. This module implements functions in PHP for opening POP/IMAP mailboxes, and is a requirement for bounce handling. Ensure that your hosting provider includes this module.

You will also need a mailbox for the bounces to be delivered to. Depending on your server configuration, this could be a local or remote POP3, or (under certain special circumstances) simply a mailbox file.

Finally, for PHP 4.2.3 and greater, "safe mode" will need to be disabled.



The PHP safe mode is an attempt to solve the shared-server security problem. It is architecturally incorrect to try to solve this problem at the PHP level, but since the alternatives at the web server and OS-levels aren't very realistic, many people, especially ISPs, use safe mode for now—<http://php.net/manual/en/features.safe-mode.php>.

## Configuring basic bounce settings

Having fulfilled the aforementioned requirements (php-imap module, available mailbox, and safe mode disabled), let's edit `config.php` to configure bounce processing:

1. Edit `config/config.php` and search for `message_envelope`.
2. From this section downwards, uncomment (remove the "#" in front of the line) and customize the following variables, so that phpList knows where to find the bounce e-mails:

```
$message_envelope = 'listbounces@yourdomain';
```

3. Change this to your bounce e-mail address. For example, `bouncybouncy@mydomain.com`. This won't replace the "from" address, but will be the "return-path" address on the e-mail headers. (The bounce address is used for replies, but the return-path address is used by e-mail systems for bounces:)

```
$bounce_protocol = 'pop';
```

4. In most cases, you will leave this unchanged. Only modify this if you don't have pop access to your bounce mailbox

```
define ("MANUALLY_PROCESS_BOUNCES", 1);
```

5. This controls whether or not we have an option in the admin sidebar to process bounces. Let's leave this unchanged for now

```
$bounce_mailbox_host = 'localhost';
```

6. Assuming we're using a POP mailbox, this refers to the mail host on which the mailbox is located. If your mail host is the same as your phpList host, then leaving this unchanged will work; else you might change it to whatever mail host your provider has set up, that is, `mail.mydomain.com`.  

```
$bounce_mailbox_user = 'popuser';
```
7. This is the mailbox username. You'll want to change this, unless your bounce user is really called `popuser`.  

```
$bounce_mailbox_password = 'password';
```
8. Likewise, you'll almost certainly want to change this from the default of `password!`  


```
$bounce_mailbox_port = "110/pop3/notls";
```
9. This specifies the protocol used (by default, unencrypted POP3) to access the mailbox. You'll want to leave this unchanged unless (a) you experience problems or (b) your provider requires transmission encryption on POP3 connections; in that case, try changing it to `110/pop3` instead.  

```
$bounce_mailbox = '/var/spool/mail/listbounces';
```
10. If you set `$bounce_protocol` above `mailbox`, then this is the location to your mailbox file on the server's filesystem. If not, you can safely ignore this one.  

```
$bounce_mailbox_purge = 1;
```
11. Sometimes, for testing purposes, you might want to process and re-process the same bounces in the mailbox. If this is the case, you'll set this value to 0, but under normal operation, it should remain at 1.  

```
$bounce_mailbox_purge_unprocessed = 1;
```
12. This (advanced) option will leave "unprocessed" bounces (those which could not be matched to a user) in the mailbox. Change this to 0 if you intend to manually review these unprocessed bounces using a mail client; or else leave this at the default of 1.  

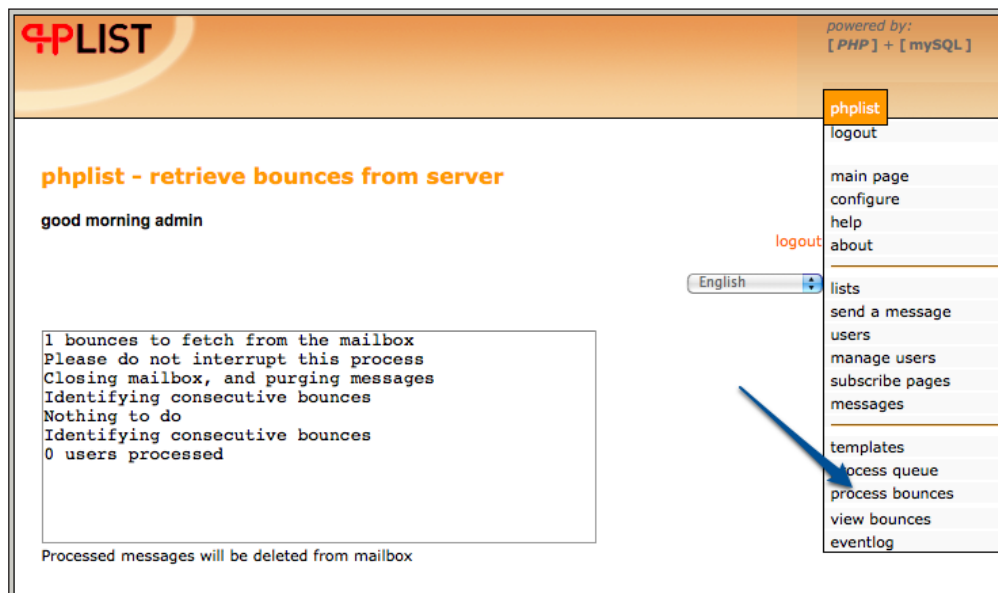
```
$bounce_unsubscribe_threshold = 5;
```
13. The "Bounce Threshold" is probably the only field which may require tweaking – this defines the number of "strikes-before-you're-out" on bounces. That is, after this many (default 5) **consecutive** bounces, a member will be automatically unsubscribed.

 In versions of phpList prior to 2.10.2, this variable was incorrectly named `$bounce_unsubscribe_treshold` (the first h of "threshold" was missing).

## Testing and manually processing bounces

Now that we've defined our bounce settings (specifically the mailbox settings), let's test our bounce-processing by sending an e-mail to our bounce mailbox (the envelope-address) using a regular e-mail program, and then using the **process bounces** item on the right-hand panel in the admin page.

We should see at least one message processed, indicating that delivery to and inspection of our mailbox is working:



Note that in the preceding example, zero users were processed. This is because phpList was (quite rightly) unable to match our test message with an existing subscriber.

Here's what the output would look like with a real bouncing subscriber:

powered by:  
[ PHP ] + [ MySQL ]

phplist

logout

main page

configure

help

about

logout

English

lists

send a message

users

manage users

subscribe pages

messages

templates

process queue

process bounces

view bounces

eventlog

**phplist - retrieve bounces from server**

good morning admin

```
0 of 1 users processed
Identifying consecutive bounces
1 users processed
```

Processed messages will be deleted from mailbox

## Reviewing bounces

Having captured information about bounces, we now want to review these bounces. Click on the **view bounces** link in the right-hand panel on the admin page to review bounces:

powered by:  
[ PHP ] + [ MySQL ]

phplist

logout

main page

configure

help

about

logout

English

lists

send a message

users

manage users

subscribe pages

messages

templates

process queue

process bounces

view bounces

eventlog

**phplist - list bounces**

good morning admin

This is the list of bounces received. In most cases they will be identified for a user and a message. If you run a live system, do not delete the bounces, because the system would not be able to identify the consecutive bounces, once more bounces come back. Currently you can only view a bounce, there is no way (yet) to override the system identification of the bounce. In some cases this may mean that something is marked as a bounce, when it was only a "Mailbox full" temporary error.

4 bounces

[ delete all unidentified (> 2 months old) | Delete all processed (> 2 months old) | Delete all | Reset bounces ]

|                   | Message                | User    | Date                |
|-------------------|------------------------|---------|---------------------|
| [ Delete   show ] | 19                     | 13      | 2010-10-09 10:20:00 |
| [ Delete   show ] | System Message Unknown | Unknown | 2010-10-09 10:20:00 |
| [ Delete   show ] | System Message Unknown | Unknown | 2010-10-09 10:10:00 |

All the processed bounces (including our earlier test messages) are listed. From this page, we can also "clean up" our bounce history, by deleting all bounces older than two months, deleting all bounces altogether, or resetting all members' bounce counts (that is, in the event of an internal mail delivery issue causing members' mail to bounce).

## Taking action on individual bounces

To view an individual bounce, click on the **show** link, as shown in the previous screenshot. We will see the details of the bounce message:

powered by: [PHP] + [MySQL]

phplist

logout

main page  
configure  
help  
about

logout

English

phplist - view a bounce

good morning admin

Possible Actions:

For user with email

Increase bouncecount with  (use negative numbers to decrease)

Mark user as unconfirmed  (so you can resend the request for confirmation)

Set user to receive text instead of HTML

Delete user

Delete this bounce and go to the next

Bounce Details:

|         |                          |
|---------|--------------------------|
| ID      | 3                        |
| Date    | 2010-10-09 10:20:00      |
| Status  | bounced list message 19  |
| Comment | 13 bouncecount increased |

lists  
send a message  
users  
manage users  
subscribe pages  
messages

templates  
process queue  
process bounces  
view bounces  
eventlog

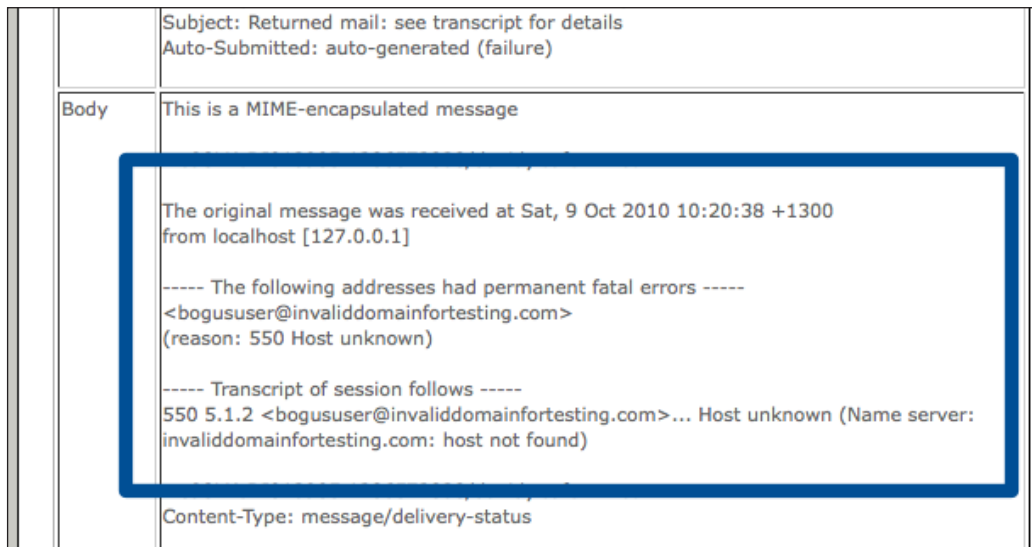
From this page, we can:

- Adjust the user's bounce count (that is, give the user more or less "strikes" before they're out)
- Mark the user as unconfirmed (stop sending him/her e-mail)
- Adjust the user's formatting preferences
- Permanently delete the user
- Delete this bounce from the database and review the next one sequentially

## Interpreting why an e-mail bounced

Although we can perform individual actions based on the bounce (deleting the user, for example), the most useful part of this page is the copy of the actual bounce message that helps us determine why this message bounced.

Scrolling down to the body of the bounce message normally yields an explanation. That is, in the following example, our message bounced because it was sent to an invalid e-mail address:



Some reasons why an e-mail might bounce are:

- Invalid e-mail address
- Temporary issue delivering e-mail to a destination server
- The recipient's mailbox may be full



## Examining users whose e-mail bounces

From the **view bounces** menu, we can click on the user number to examine a particular user whose e-mail bounces:

powered by: [ PHP ] + [ mySQL ]

phplist

logout

main page

configure

help

about

logout

English

lists

send a message

users

manage users

subscribe pages

messages

templates

process queue

process bounces

view bounces

eventlog

**phplist - list bounces**

good morning admin

This is the list of bounces received. In most cases they will be identified for a user and a message. If you run a live system, do not delete the bounces, because the system would not be able to identify the consecutive bounces, once more bounces come back. Currently you can only view a bounce, there is no way (yet) to override the system identification of the bounce. In some cases this may mean that something is marked as a bounce, when it was only a "Mailbox full" temporary error.

4 bounces  
[ delete all unidentified (> 2 months old) | Delete all processed (> 2 months old) | Delete all | Reset bounces ]

| Message                                  | User | Date                |
|--|------|---------------------|
| [ Delete   show ] 19                     | 13   | 2010-10-09 10:20:00 |
| [ Delete   show ] System Message Unknown |      | 2010-10-09 10:20:00 |
| [ Delete   show ] System Message Unknown |      | 2010-10-09 10:10:00 |

Examining the user, in this context, tells us the following:

- How many consecutive bounces we've received from this user
- Whether this user has been unconfirmed due to exceeding the consecutive bounce threshold

powered by: [ PHP ] + [ mySQL ]

phplist

logout

main page

configure

help

about

logout

English

lists

send a message

users

manage users

subscribe pages

messages

templates

process queue

process bounces

view bounces

eventlog

**phplist - details of a user**

good morning admin

back to the list of users

Delete will delete user and all listmemberships

delete

boguser@invalidomainfortesting.com update page unsubscribe page history

User Details

|                                |                                    |
|--------------------------------|------------------------------------|
| ID                             | 13                                 |
| Email                          | boguser@invalidomainfortesting.com |
| Is this user confirmed (1/0)   | 1                                  |
| Is this user blacklisted       | 0                                  |
| Number of bounces on this user | 1                                  |

## Reviewing unconfirmation actions in the eventlog

When a user's consecutive bounce count exceeds the defined threshold, they will be automatically unconfirmed. You can see this action in the **eventlog**:

The screenshot shows the phplist administration interface. At the top, it says 'powered by: [PHP] + [mysql]'. The main heading is 'phplist - log of events' with the user 'good morning admin' logged in. Below this, there's a description of the event log and a table of events. The first event is highlighted, showing a user marked as unconfirmed due to exceeding the bounce threshold. A blue arrow points to the 'eventlog' link in the right-hand navigation menu.

| Events del | page               | date                | message  |
|------------|--------------------|---------------------|--|
| 63         | del processbounces | 2010-10-09 10:54:11 | User 13 has consecutive bounces (5) over treshold, user marked unconfirmed |

## Summary

In this chapter, you've learned about how to configure and manage e-mail bounce handling, including:

- Basic bounce settings
- Manually processing bounces
- Reviewing and acting on bounces

In *Chapter 9, Advanced Features*, we'll cover more advanced bounce processing, as well as configuring automated bounce handling. In the following chapter, we'll look at integrating phplist with popular publishing platforms, such as Blogger, Drupal, Joomla, and WordPress.



# 4

## Setting up phpList with Popular Publishing Platforms

In *Chapter 2, Setting up Subscriber Forms*, you learned how to create phpList subscribe pages and how to add these pages to other platforms by copying and pasting the form code into an HTML box, on platforms such as Blogger.

There are some platforms that offer deeper integration with phpList and we'll cover three of the most popular in this chapter, namely:

- Integrating phpList with Drupal
- Integrating phpList with WordPress
- Integrating phpList with Facebook

These platforms were chosen for this chapter because of the phpList integration already available to them using open source modules (Drupal) and plugins (WordPress), or because of the platform's own extensibility or popularity (Facebook).

### Integrating phpList with Drupal

Drupal is described as a "community publishing system". It has a mature, modular, well-documented "core" of features, providing general content management functions such as content, user, and system management.

Drupal is then extensible through modules, which add features like e-commerce, advanced taxonomy, SEO optimizations, and so on. In our case, we're going to install and configure a module to add phpList integration.

## Prerequisites

For this section, we'll make the following assumptions:

- We already have a working instance of Drupal (version 6).
- We are hosting our Drupal site and our phpList site on the same web server and with the same URL base. That is, our Drupal site is accessible at <http://yoursite.com> and our phpList installation is accessible at <http://yoursite.com/lists/>.

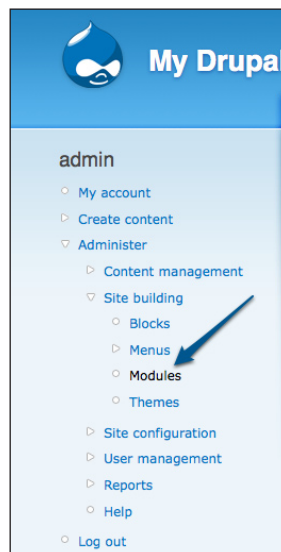


We chose to document the Drupal-phpList integration using Drupal 6, even though Drupal 7 has recently been released. This is because (a) the phpList module for Drupal 7 is still marked as "development" and (b) Drupal 6 has been the official stable version for three years and has a more familiar interface than 7 at this time.

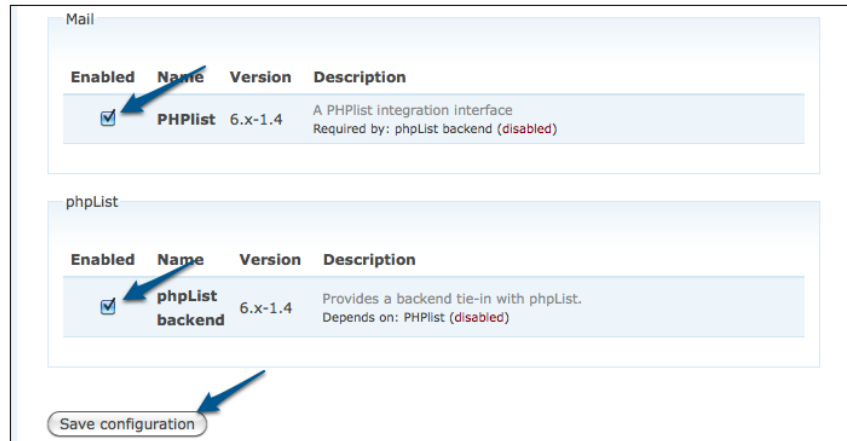
However, the following method described for Drupal 6 will work on Drupal 7.

## Installing and configuring the phpList integration module

Go to <http://drupal.org/project/phpList> and download the latest stable version of the module for Drupal 6.x. Unpack the `tar.gz` file and you should have a folder called `phpList` inside. Upload this folder to your Drupal installation's `modules` directory and then navigate to **Administer** | **Site building** | **Modules**:

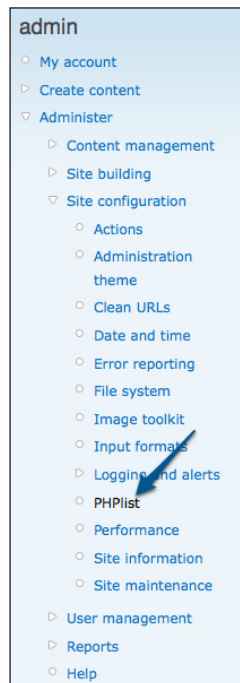


At the bottom of the modules list, you'll find the **Mail** and **phpList** headings with a single phpList module under each. Check both and click on **Save configuration**:



## External phpList configuration

Navigate to **Administer | Site configuration | PHPList** to set up the database credentials and other options required for the integration:



You are prompted for your phpList database details. Enter your database host, database name, username, and password. Unless you've done a non-standard installation of phpList, the default entries for prefix and user table prefix will already be correct.

Under **PHPList URL**, enter the URL to your phpList installation. Because we are using phpList and Drupal at the same base URL, we set **/lists/** (with a trailing slash) as our **PHPList URL**:

Home > Administer > Site configuration

PHPList **Settings** List access

External PHPList configuration

**Database host:**  
localhost  
Host name for the PHPList database

**Database name:**  
my-database-name  
Name of the PHPList database

**Database user:**  
my-database-user  
User name for the PHPList database

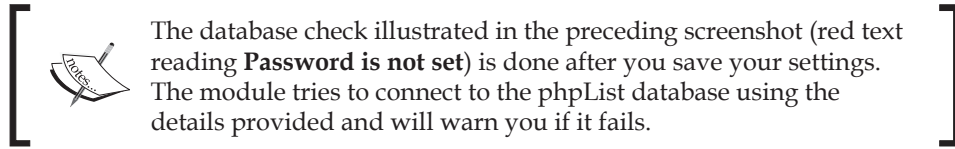
**Database password:**  
.....  
Password for the PHPList database. Characters not allowed: # / ?

Password is not set

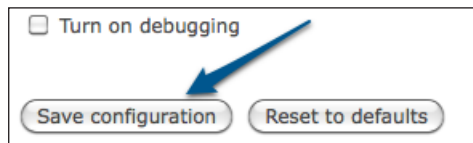
**Table prefix:**  
phplist\_  
Prefix for the database tables

**User table prefix:**  
phplist\_user\_  
Prefix for the user table. Blank = same as table prefix

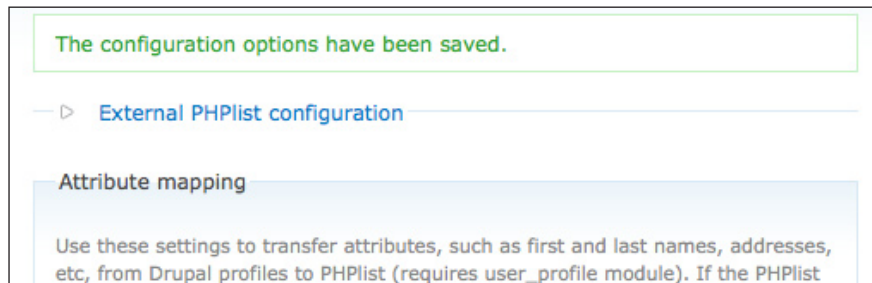
**PHPList URL: \***  
/lists  
The absolute path or URL of your PHPList front page, for example: /lists/ or http://lists.mydomain.com/



Scroll to the bottom, ignoring the other options for now, and click on **Save configuration**:



If the database connection test was successful, the **External PHPList configuration** options will be collapsed into a single, clickable field, hiding them from normal view, as you configure the remaining options:



## Attribute mapping

The module can auto-create attributes in phpList to match any attributes created in your Drupal instance. For example, you may ask your Drupal members to enter demographic information when registering and this mapping would allow these details to be transferred to the phpList.



Note that the module warns you that this mapping only works well with textline attributes and not select attributes or radio buttons. Use of this feature is not covered in this chapter, as it requires advanced pre-configuration of your Drupal instance:

**Attribute mapping**

Use these settings to transfer attributes, such as first and last names, addresses, etc, from Drupal profiles to PHPList (requires user\_profile module). If the PHPList attributes do not already exist, they will be created. Data will be transferred from Drupal profiles and overwrite the corresponding PHPList attribute data (exception: if the Drupal profile field is blank the PHPList attribute is left alone).

Note that this works well with PHPList textline attributes, it may work with dates and checkbox attributes, and it **will not work** with select attributes, radio buttons, and most other attribute types. Please test carefully on a few users before proceeding to synchronise all users!

Use the SYNCHRONISE NOW link to refresh all existing accounts.

**Map non-system roles**  
Create PHPList attributes for each Drupal non-system user role (i.e., anything other than anonymous user & authenticated user) which is appropriately checked or un-checked for each user

## **My Account – My Newsletters options**

The module adds a tab to your **My Account** page in Drupal, allowing a user to modify their list subscriptions. You can customize several aspects of this page, including:

- The text to show on the **My Newsletters** tab
- The title of the **My Newsletters** tab itself (that is, you may want it to read "My Subscriptions")
- Whether or not to show mailing list descriptions (the list description you entered in phpList)
- Whether to offer the user the choice of HTML or Text e-mail and which one should be the default

My Account - My Newsletters options

**Text to show on the user/edit page:**

Thank you for subscribing to our newsletters. We promise to respect your privacy, and not to share your details with any third parties!

**Text to show in the user/edit screen as the category title for subscribe/unsubscribe to PHPlist newsletters :**

My newsletters

Show mailing list descriptions

Hide email format selection  
Applies to new user registration and My Account

**Default mail format:**

html

If not showing the email format box (in the block or My Account), default to this format

## Registration Page options

The module can optionally display a subscription page on the registration page that Drupal shows to prospective members as they register. This would allow new members to select their subscription preferences at the same time as they create their Drupal account. By default, this is disabled:

Registration Page options

Show subscription checkboxes in user registration form

Show mailing list descriptions

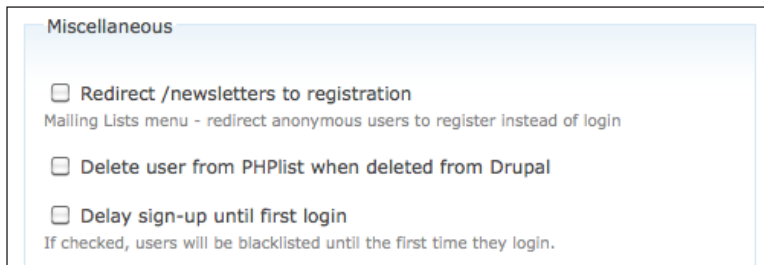
**Text to show on the registration form:**

These are mailinglists which are offered FREE to our members. Please check any you want to subscribe to.

## Miscellaneous

There are three unclassified options offered under the **Miscellaneous** heading:

- **Redirect /newsletters to registration:** Normally, if an unauthenticated user tries to go to the **My newsletters** URL in Drupal, they would be presented with a login screen to authenticate. This option replaces this default behavior and, instead, presents them immediately with the registration page.
- **Delete user from PHPList when deleted from Drupal:** To keep your phpList database tidy, you may want to automatically delete phpList's version of a user when their Drupal account is deleted.
- **Delay sign-up until first login:** This option will leave a subscribed user in a blacklisted state in phpList until they've logged into Drupal for the first time. This is to prevent a malicious user from filling out the registration form on behalf of an unsuspecting victim and having that victim immediately subscribed to your mailing lists.

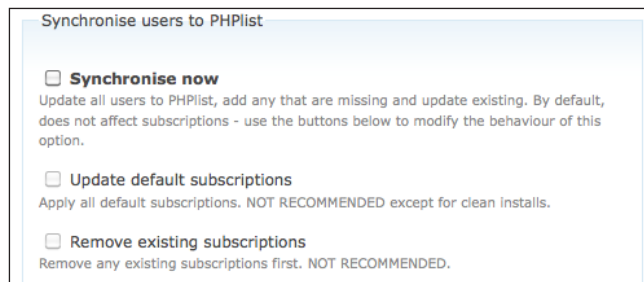


The screenshot shows a light blue box titled "Miscellaneous" containing three settings, each with an unchecked checkbox:

- Redirect /newsletters to registration**  
Mailing Lists menu - redirect anonymous users to register instead of login
- Delete user from PHPList when deleted from Drupal**
- Delay sign-up until first login**  
If checked, users will be blacklisted until the first time they login.

## Synchronize users to phpList

Normally, synchronization between Drupal and phpList happens when a user initiates the change (that is, they register or they change their subscription preferences). However, as this is the first time the phpList integration has been introduced, you may want to force an immediate synchronization of all existing Drupal users, by checking the **Synchronise now** checkbox:



The screenshot shows a light blue box titled "Synchronise users to PHPList" containing three settings, each with an unchecked checkbox:

- Synchronise now**  
Update all users to PHPList, add any that are missing and update existing. By default, does not affect subscriptions - use the buttons below to modify the behaviour of this option.
- Update default subscriptions**  
Apply all default subscriptions. NOT RECOMMENDED except for clean installs.
- Remove existing subscriptions**  
Remove any existing subscriptions first. NOT RECOMMENDED.

The lower two checkboxes control whether the subscription information (as opposed to simply the user information) will be altered in the synchronization process. You would only conceivably use the **Update default subscriptions** and **Remove existing subscriptions** checkboxes if you were working on a clean install without existing users.

## Turn on debugging

Enabling the **Turn on debugging** checkbox will add some debugging status boxes to help identify problems. This should be left disabled unless required.

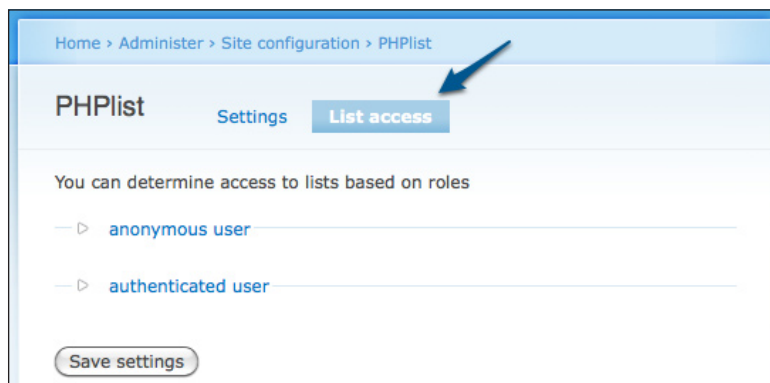


A screenshot of a configuration form. At the top, there is a checkbox labeled "Turn on debugging" which is currently unchecked. Below the checkbox are two buttons: "Save configuration" and "Reset to defaults".

## Configuring list access

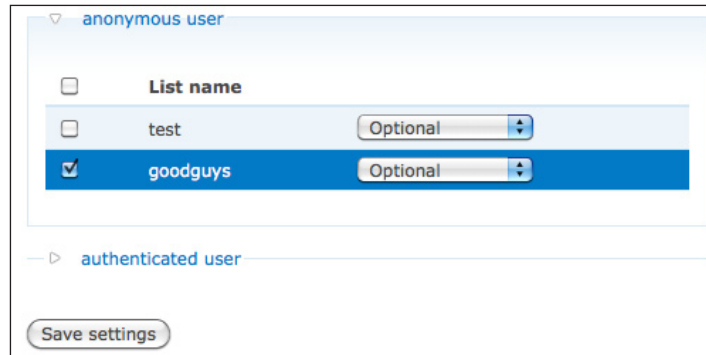
Now that the global integration options are configured, let's configure which Drupal user roles may sign up to which phpList lists.

Under the **Administer** | **Site Configuration** | **PHPlist** page, click on the **List access** tab. You're prompted to select user access based on roles and the two default Drupal roles (**anonymous user** and **authenticated user**) are displayed:



A screenshot of the PHPlist configuration page. The breadcrumb trail at the top reads "Home > Administer > Site configuration > PHPlist". Below this, there are two tabs: "Settings" and "List access", with "List access" being the active tab. A blue arrow points to the "List access" tab. The main content area shows the heading "You can determine access to lists based on roles" followed by two expandable sections: "anonymous user" and "authenticated user". At the bottom of the page is a "Save settings" button.

Expand each role and select (with the checkbox) all the lists that this role is allowed to subscribe to and click on **Save settings**:



| <input type="checkbox"/>            | List name | Optional |
|-------------------------------------|-----------|----------|
| <input type="checkbox"/>            | test      | Optional |
| <input checked="" type="checkbox"/> | goodguys  | Optional |

Save settings



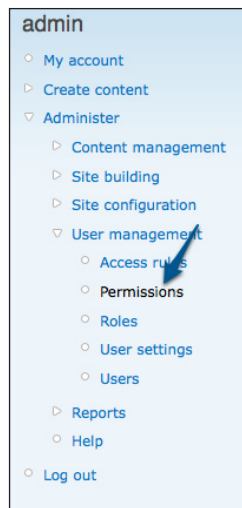
Remember that Drupal considers unauthenticated users as anonymous and logged-in users as authenticated. If you make some lists accessible to the anonymous user role, but not the authenticated user role, then the logged-in users will be unable to manage their subscription.

Any list marked as optional is available for subscription and any marked as auto-subscribe is subscribed to by default.

## Configuring Drupal permissions

Now that we've carried out all the configuration inside the phpList module, we also need to grant permissions to the various phpList functions to user roles in Drupal.

Navigate to **Administer | User management | Permissions** to assign these permissions:



You're shown a table of all the available permissions in Drupal, with a column for each user role (by default, **anonymous** and **authenticated**).

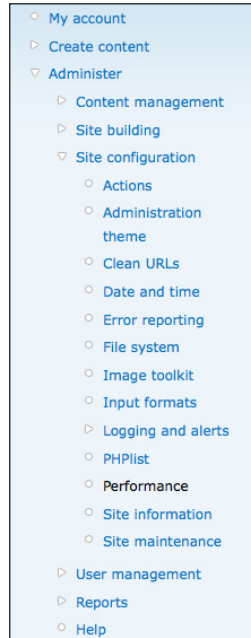
Scroll down to the **phplist module** section and grant **access lists** permission to all the user roles that you want to be able to subscribe to lists. Leave the **administer PHPlist** and **manage subscriptions** (of other users) unchecked as these are administrator-only functions:

| phplist module       |                                     |                                     |
|----------------------|-------------------------------------|-------------------------------------|
| access lists         | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| administer PHPlist   | <input type="checkbox"/>            | <input type="checkbox"/>            |
| manage subscriptions | <input type="checkbox"/>            | <input type="checkbox"/>            |

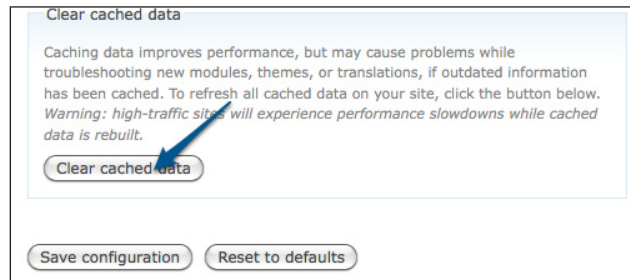
## Clearing Drupal's cache

We need to clear Drupal's cache, so that our new newsletters tab will show up in our users' profile management page.

Navigate to **Administer** | **Site configuration** | **Performance**:

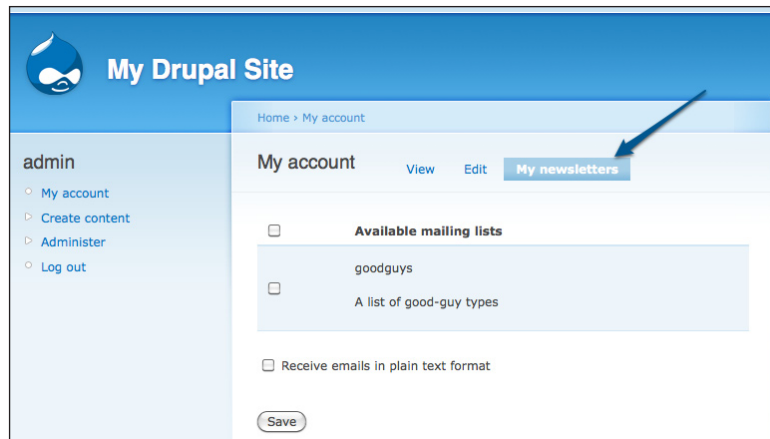


Scroll to the bottom of the **Performance** page and click on the **Clear cached data** button to clear the cache:



## Confirming access to My newsletters

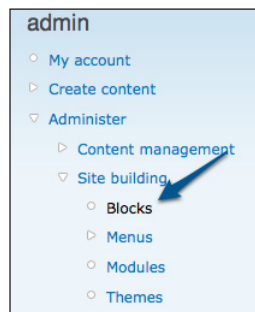
Having all the requirements in place, click on your account (**My account**) and then on the **My newsletters** tab. You will see the phpList lists that you've made available to the authenticated users role:



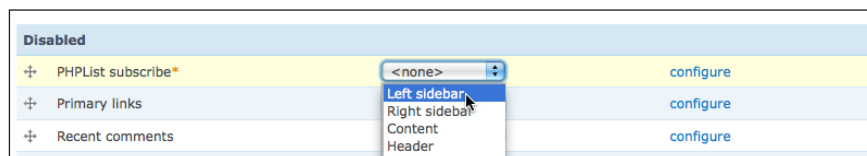
## Adding the phpList subscribe block

You might want to use the "block" that the phpList module provides. This will put a subscription form (or a link to your current subscriptions) on the sidebar of every Drupal page.

1. Navigate to **Administer** | **Site building** | **Blocks**:



2. Here a list of all currently active blocks is displayed section-by-section followed by a list of currently disabled blocks. Change the **PHPList subscribe** block to **Left sidebar** to activate it and then click on **Save blocks** at the bottom of the page to save your changes:





3. Having reloaded the page after saving, click on the configure link next to the **PHPList subscribe** block to confirm the configuration:

| Block               | Region       | Operations                |
|---------------------|--------------|---------------------------|
| <b>Left sidebar</b> |              |                           |
| + Navigation        | Left sidebar | <a href="#">configure</a> |
| + User login        | Left sidebar | <a href="#">configure</a> |
| + PHPList subscribe | Left sidebar | <a href="#">configure</a> |

4. You can make any phpList-specific changes under the **Block specific settings** section, including whether or not to include an e-mail confirmation field, whether to include list descriptions, and the option of HTML/Text format:

Block specific settings

**Block title:**  
Mailinglists  
Override the default title for the block. Use <none> to display no title, or leave blank

**Block text:**  
We have several high-quality mailinglists available for FREE  
Explanatory text that will be displayed above the subscription form.


**Input format**

Show email confirmation field

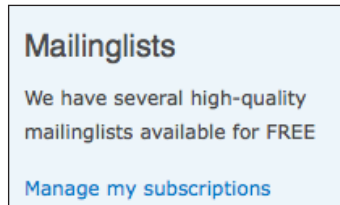
Show mailing list descriptions on the PHPList subscription block

Show email format box

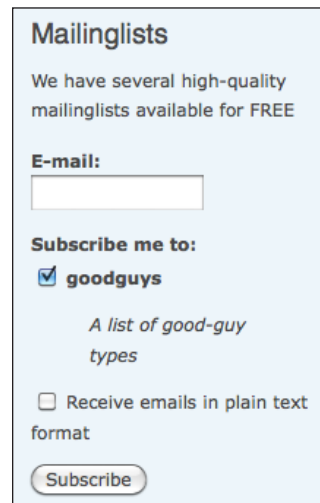
**Size of email box(es):**  
16

[  Drupal also gives you flexible control over the display of the block, including which users may see it, which pages it is displayed on, and so on. ]

- Having saved your block settings, the phpList subscribe block is now visible in the left-hand navigation panel, for an authenticated user, as follows:

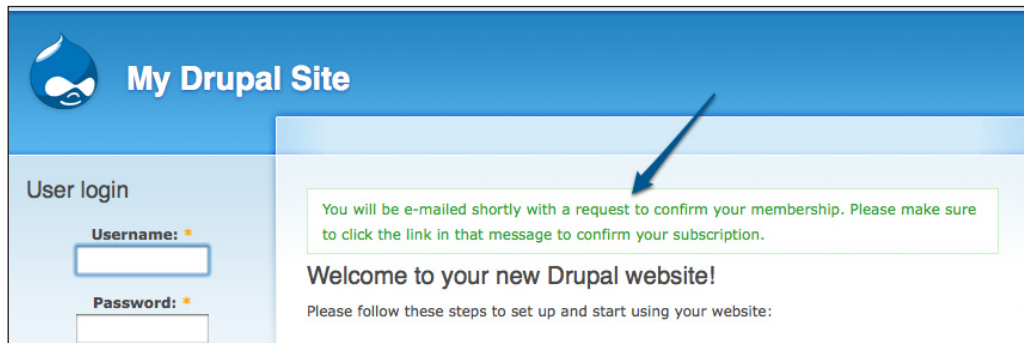


- It is also visible for unauthenticated (anonymous) users, as follows:



The screenshot shows a light blue box with the title "Mailinglists". Below the title, it says "We have several high-quality mailinglists available for FREE". There is an "E-mail:" label followed by a text input field. Below that is a "Subscribe me to:" label, followed by a checked checkbox and the text "goodguys". Underneath, it says "A list of good-guy types". There is also an unchecked checkbox labeled "Receive emails in plain text format". At the bottom of the box is a "Subscribe" button.

- Once the user is done entering an e-mail and making a list selection, they will receive the following message:



## Troubleshooting the phpList subscribe block

There are a few prerequisites for correct functioning of the subscribe block:

- The **phpList URL** you entered in the module configuration must be correct
- The default phpList subscribe page must have no mandatory attributes
- The phpList language must be **English** (the module looks for the text **Thank you for subscribing** to indicate success)

## Integrating phpList with WordPress

*"WordPress is an open source blog tool and publishing platform powered by PHP and MySQL. It's often customized into a Content Management System (CMS). It has many features including a plug-in architecture and a template system. WordPress is used by over 13% of the 1,000,000 biggest websites".*  
Source: <http://en.wikipedia.org/wiki/WordPress>

### Prerequisites

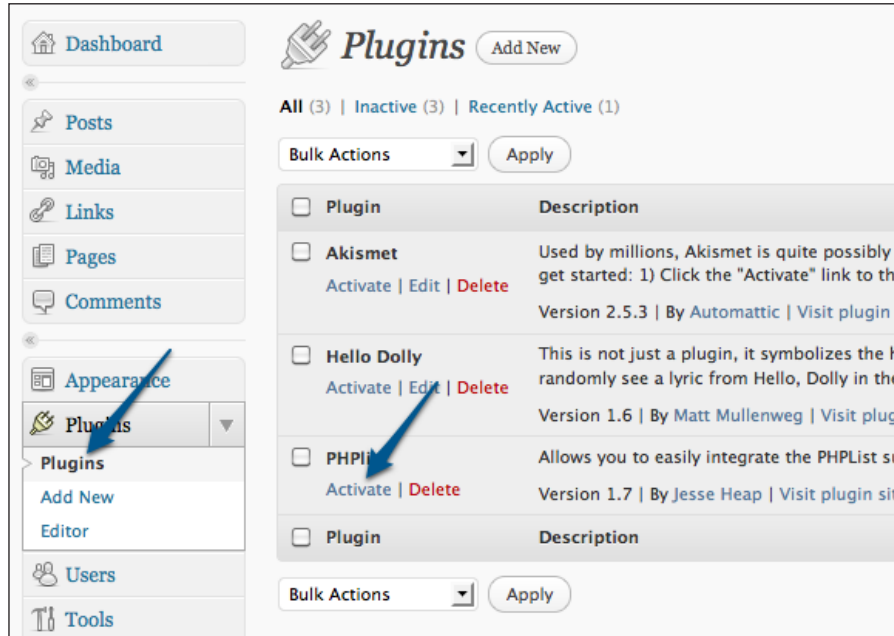
For this section, we'll make the following assumptions:

- We already have a working instance of WordPress (version 3.x)
- Our phpList site is accessible through HTTP / HTTPS from our WordPress site

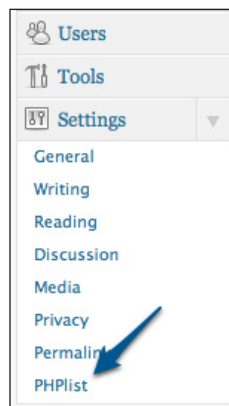
## Installing and configuring the phpList Integration plugin

Download the latest version of Jesse Heap's phpList Integration plugin from <http://wordpress.org/extend/plugins/phplist-form-integration/>, unpack it, and upload the contents to your `wp-content/plugins/` directory in WordPress.

Activate the plugin from within your WordPress dashboard:



Under the **Settings** menu, click on the new **PHPlist** link to configure the plugin:



## General Settings

Under the **General Settings** heading, enter the URL to your phpList installation, as well as an admin username/password combination. Enter the ID and name of at least one list that you want to allow your WordPress users to subscribe to:

**General Settings**

PHPList URL:

PHPList Admin Login:  Enter PHPList Admin Login

PHPList Admin Password:  Enter PHPList Admin Password

PHPList List Information:  Enter PHPList ID  Enter Name of List

For each list enter the list number and name of the list in the above table. [See help](#) for more info.

Skip Confirmation Email (Check to bypass confirmation email)



### Why does the plugin require my admin login and password?

The admin login and password are used to bypass the confirmation e-mail that would normally be sent to a subscriber. Effectively, the plugin "logs into" phpList as the administrator and then subscribes the user, bypassing confirmation.

If you don't want to bypass confirmation e-mails, then you don't need to enter your username and password.

## Form Settings

The plugin will work with this section unmodified. However, let's imagine that we also want to capture the subscriber's name. We already have an attribute in phpList called **first name**, so change the first field label to **First Name** and the **Text Field ID** to **first name** (the same as our phpList attribute name):

**Form Settings**

Use these settings to **OPTIONALLY** configure your form. Currently the form supports two text fields. The first field supports any text label and should correspond to a text field you capture in PHPList. Most people capture NAME in this field. The second field is the **required** email field.

| Show on Form?                       | Text Field Label Name | Text Field ID | Text Field Size                 | Text Field Max Size             | Required Field?                     |
|-------------------------------------|-----------------------|---------------|---------------------------------|---------------------------------|-------------------------------------|
| <input checked="" type="checkbox"/> | First Name            | first name    | <input type="text" value="30"/> | <input type="text" value="50"/> | <input type="checkbox"/>            |
| <input checked="" type="checkbox"/> | Email                 | email         | <input type="text" value="20"/> | <input type="text" value="50"/> | <input checked="" type="checkbox"/> |

Check to show held on form Enter Text Label (i.e. Name) Enter Text ID - see help for more information Enter size of text field Enter max size of text field Check to make field required

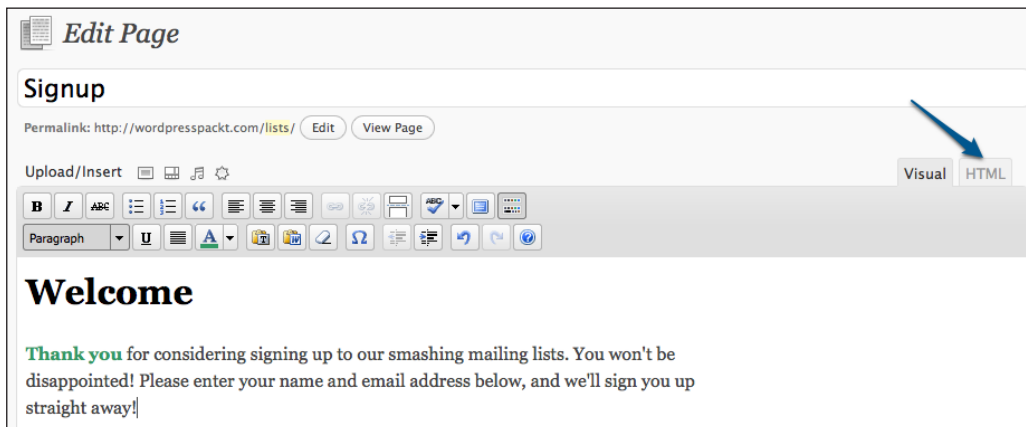
Use default CSS for subscriber form. (Uncheck to use your own CSS for the form)

## Adding a phpList Integration page

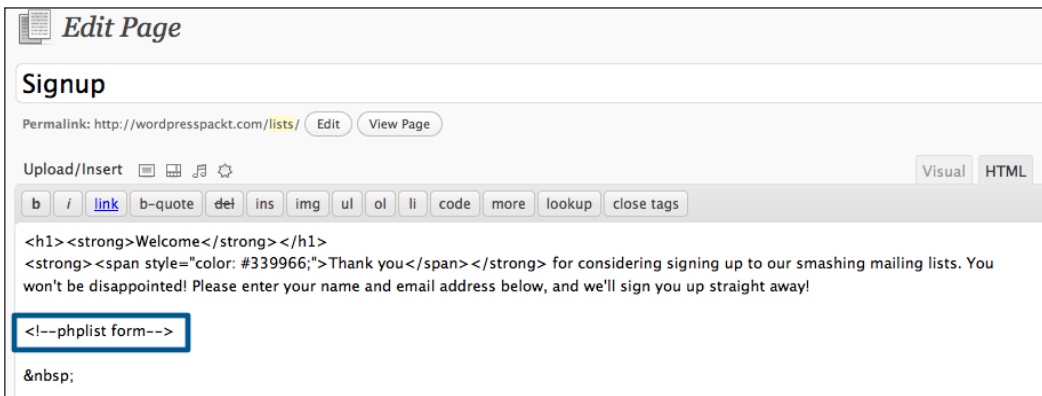
The plugin will replace the HTML comment `<!--phplist form-->` with the generated phpList form.

Let's say we wanted our phpList form to show up at `http://ourblog.com/signup`.

Create a new WordPress page called **Signup**, add the content you want to be displayed, and then click on the **HTML** tab to edit the HTML source:



You will see the HTML source of your page displayed. Insert the text `<!--phplist form-->` where you want the form to be displayed and save the page:

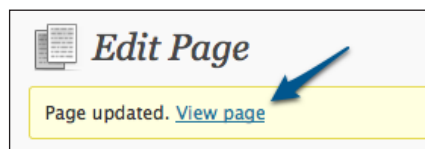




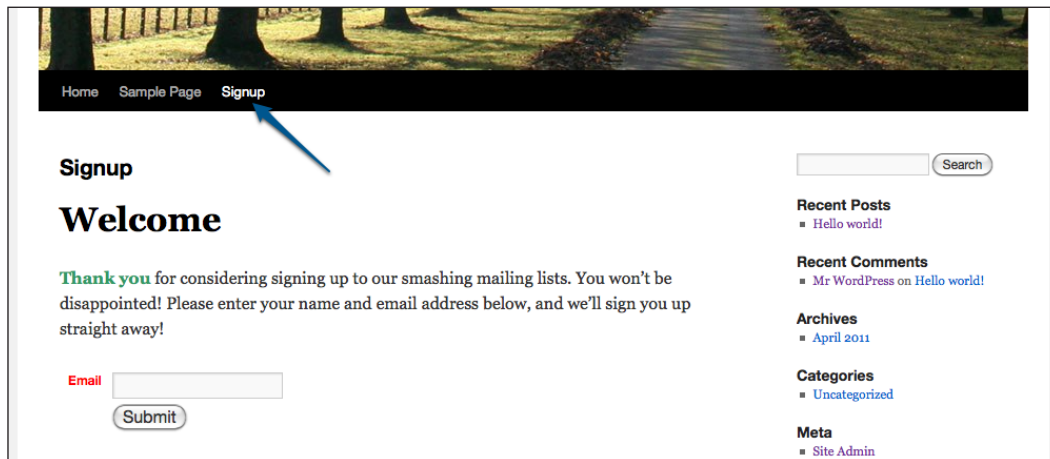
### HTML comments

The "`<!--some text-->`" syntax designates an HTML comment, which is not displayed when the HTML is processed by the browser / viewer. This means that you won't see your comment when you view your page in Visual mode.

Once the page has been updated, click on the **View page** link to display the page in WordPress:



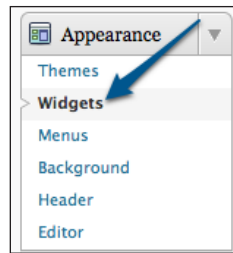
The subscribe form will be inserted in the page at the location where you added the comment:



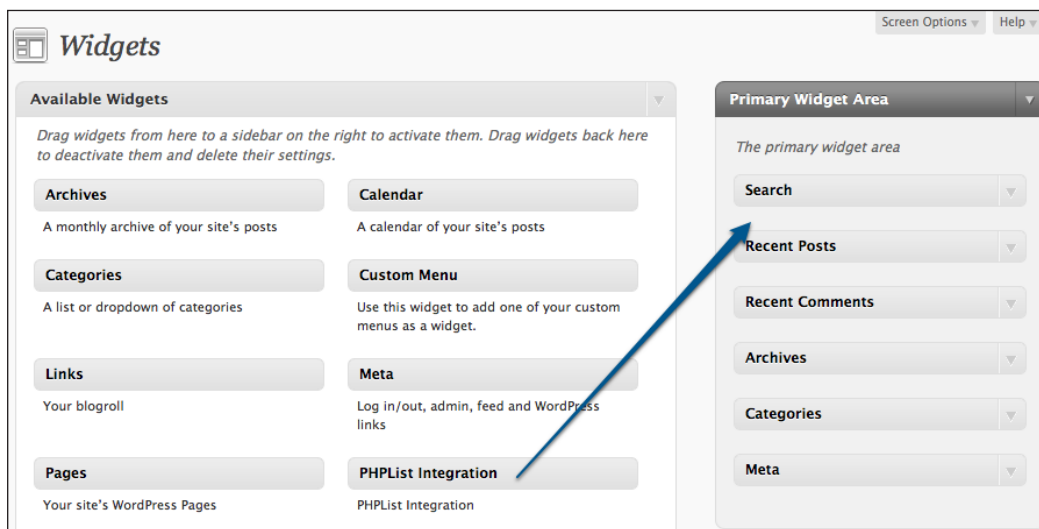
## Adding a phpList Integration widget

Instead of a dedicated page to sign up new subscribers, you may want to use a sidebar widget instead, so that the subscription options can show up on multiple pages on your WordPress site.

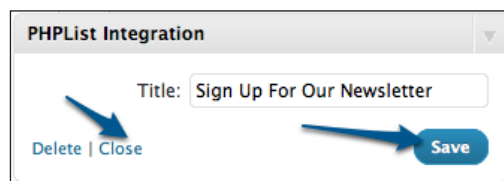
To add the phpList integration widget, go to your WordPress site's **Appearance** option and go to the **Widgets** page:



Drag the **PHPList Integration** widget to your preferred widget location. (These vary depending on your theme):

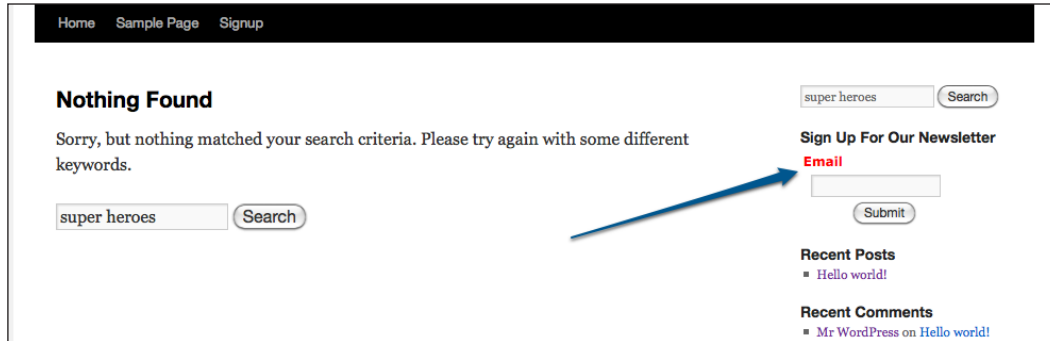


You can change the **Title** of the widget before you click on **Close** to finish:





Now that you've added the **PHPList Integration** widget to the widget area, your sign up form will be displayed on all WordPress pages, which include that widget area:



## Integrating phpList with Facebook

Facebook encourages businesses to create "Pages" that Facebook members can then "like" (by clicking the **Like** button on the page).

Once a member has "liked" a page, future updates on that page are displayed on the member's news feed. Page owners can also customize the display of their page — that is, pages can display exclusive content to fans who've already "liked" the page.

Facebook pages are an easy way to distribute news to your fans, including special offers, and product updates, and drive traffic back to your own website.

## Prerequisites

For this section, we'll make the following assumptions:

- We already have a Facebook account
- We already have a Facebook page
- We already have the Facebook developer's application installed (<http://www.facebook.com/developers>)

## Preparing phpList

Facebook allows us to include any content on our page in an iframe with a maximum width of 520 px. phpList's default theming won't fit in 520 px and so it will either get cut off and look odd, or will invoke awkward scroll bars in the middle of the page.

To resolve this, we'll need to change the styling of phpList. Note that this will also affect how the public URLs to our phpList site are displayed outside of Facebook (that is, the default sign up page).

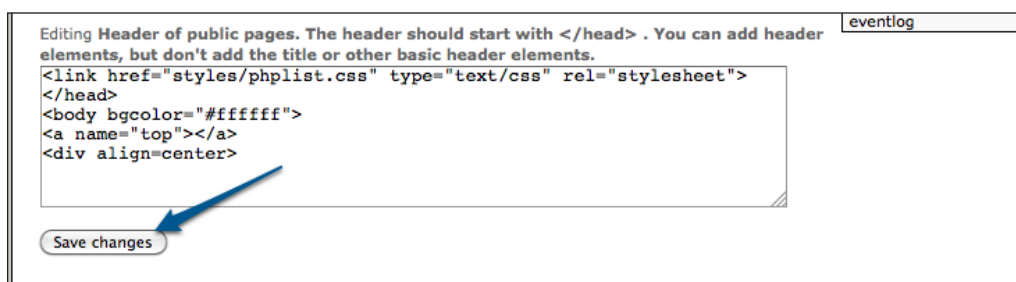
Navigate to the configure page in phpList's admin interface, using the right-hand navigation panel:



Scroll down to the section starting with **Header of public pages** and click on **edit**. This is the HTML code that is added to the top of every public (not admin) page.

```
edit Header of public pages. The header should start with </head> . You can add header
elements, but don't add the title or other basic header elements.
<link href="styles/phplist.css" type="text/css" rel="stylesheet">
</head>
<body bgcolor="#ffffff" background="images/bg.png">
<a name="top"></a>
<div align=center>
<table cellspacing=0 cellpadding=0 width=710 border=0>
<tr>
```

In the following example, we've removed the background image, the HTML table, and images, but left the CSS styling unchanged:



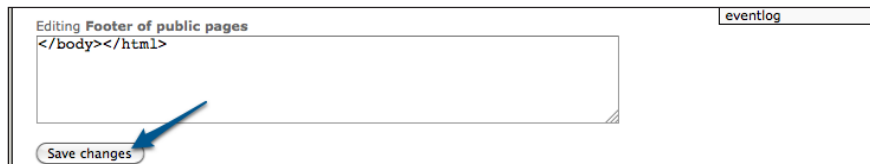
Having saved your changes, edit the next section labeled **Footer of public pages** and make the corresponding changes:

```
edit Header of public pages. The header should start with </head> . You can add header elements, but don't add the title or other basic header elements.
<link href="styles/phplist.css" type="text/css" rel="stylesheet">
</head>
<body bgcolor="#ffffff">
<a name="top"></a>
<div align=center>
```

edit Footer of public pages

Remember that the actual content that the user sees will be "sandwiched" between the header and footer. This means that if you open a tag in the header, you need to make sure it's closed again in the footer.

Again in this example, we just closed the HTML body tag in the footer:



Having changed the header and footer of the public pages, browse to your default public page (<http://your-site.com/lists/>, for example) to see how the page looks:

Subscribe to one or more of our newsletters using the form below  
required field

Email

Confirm your email address

I prefer to receive emails in HTML format

Drupal

first name

last name

hometown

Country

Please select the newsletters you want to sign up to:

- goodguys**  
A list of good-guy types
  - test**  
List for testing.

[Unsubscribe](#)

POWERED BY

Note that there is hardly any styling now, but that there are no fixed-width elements which will cause issues with our iframe. Tweaking the design of using the public header and footer code is a task left to the reader.

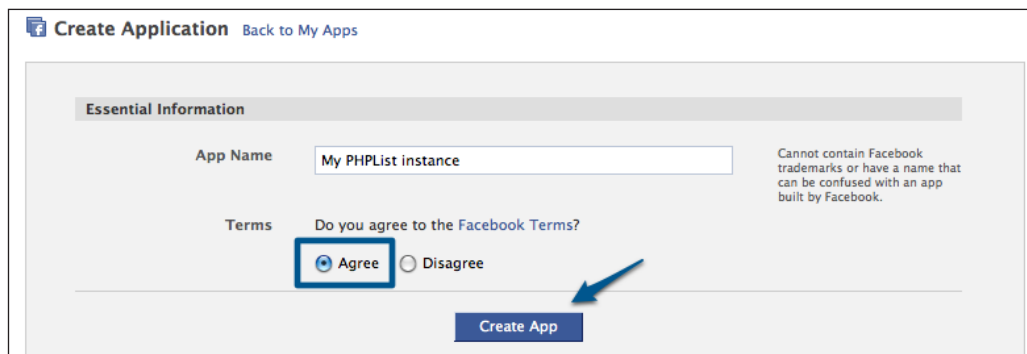
## Creating the Facebook app

Before we can add phpList to our Facebook page, we need to create an App.

From <http://www.facebook.com/developers/>, click on the **Set Up New App** button:



Click the radio button to indicate that you agree to the Facebook terms and then click on **Create App**:



Prove you're a human by completing the CAPTCHA:

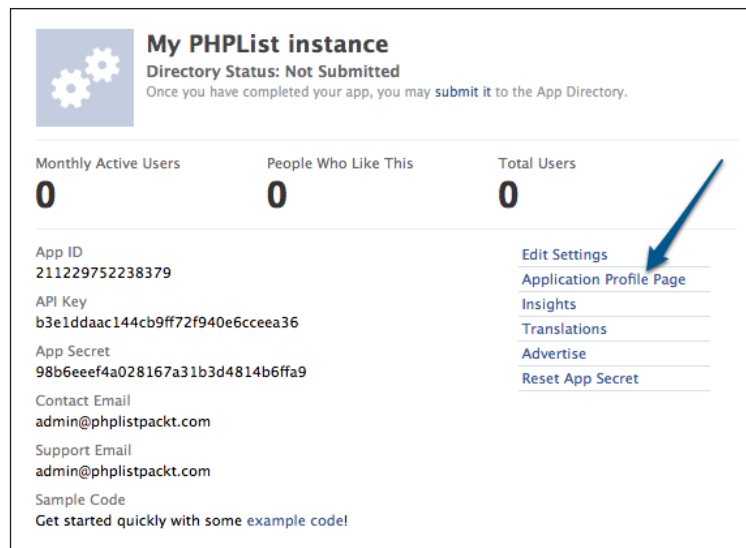


You're taken to the edit page for your new app. Under the **Facebook Integration** section, enter at least the following details:

- **Iframe Size: Auto-resize** (we don't want scrollbars)
- **Tab Name** (what the tab will be called when your page is displayed)
- **Tab URL** (the full URL to the page you want loaded in your iframe. Generally, this would be a phpList subscribe page)

The screenshot shows the Facebook Integration settings page. The left sidebar has 'Facebook Integration' selected. The main content area is divided into sections: 'Core Settings' (Application ID: 211229752238379, Application Secret: 98b6eeef4a028167a31b3d4814b6ffa9), 'Canvas' (Canvas Page: http://apps.facebook.com/[ ]/, Canvas URL, Secure Canvas URL, IFrame Size:  Auto-resize, Bookmark URL), 'Discovery' (Social Discovery:  Enabled), and 'Page Tabs' (Tab Name: Mailing Lists, Tab URL: http://phplistpackt.com/lists/?p=subscribe, Secure Tab URL, Edit URL). A blue arrow points to the 'Save Changes' button at the bottom.

Once you've saved your changes, you're taken to your list of applications. Click on **Application Profile Page** to see the application as other Facebook users would see it (as opposed to the developer):



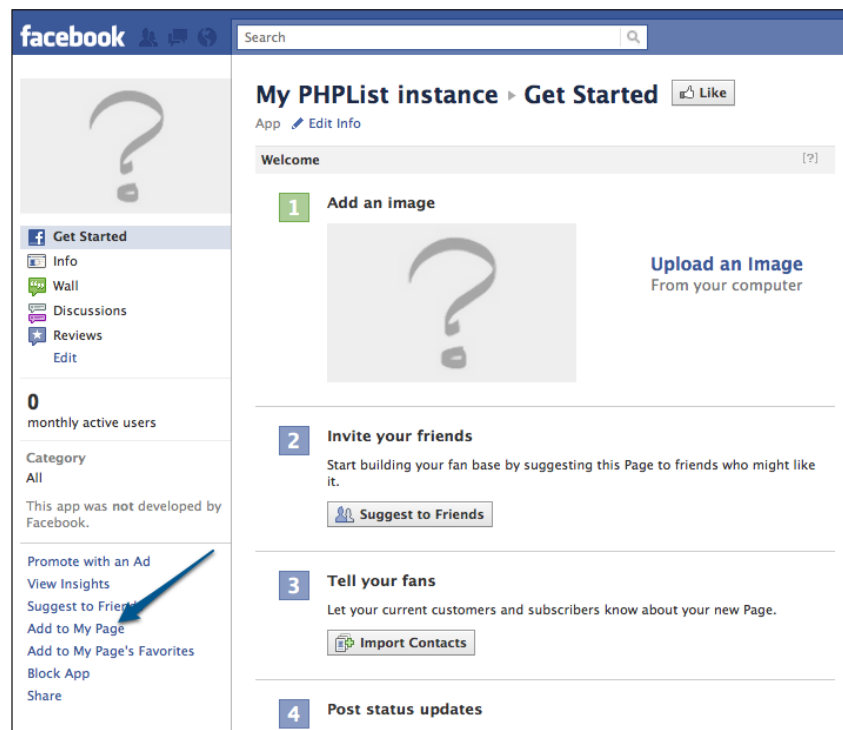
**My PHPList instance**  
Directory Status: Not Submitted  
Once you have completed your app, you may [submit it](#) to the App Directory.

|                      |                      |             |
|----------------------|----------------------|-------------|
| Monthly Active Users | People Who Like This | Total Users |
| <b>0</b>             | <b>0</b>             | <b>0</b>    |

App ID: 211229752238379  
API Key: b3e1ddaac144cb9ff72f940e6ccea36  
App Secret: 98b6eeef4a028167a31b3d4814b6ffa9  
Contact Email: admin@phplistpackt.com  
Support Email: admin@phplistpackt.com  
Sample Code: Get started quickly with some example code!

- Edit Settings
- Application Profile Page
- Insights
- Translations
- Advertise
- Reset App Secret

On the application's profile page, click on **Add to My Page** to add this application to your Facebook page:




facebook Search

**My PHPList instance** ▶ Get Started [Like](#)

App [Edit Info](#)

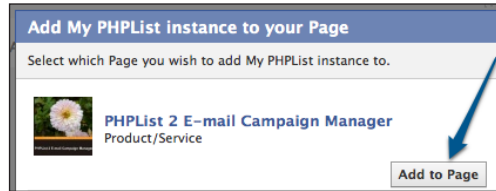
Welcome [?]

- 1 Add an image**  
  
[Upload an Image](#)  
From your computer
- 2 Invite your friends**  
Start building your fan base by suggesting this Page to friends who might like it.  
[Suggest to Friends](#)
- 3 Tell your fans**  
Let your current customers and subscribers know about your new Page.  
[Import Contacts](#)
- 4 Post status updates**

**0** monthly active users  
Category: All  
This app was not developed by Facebook.

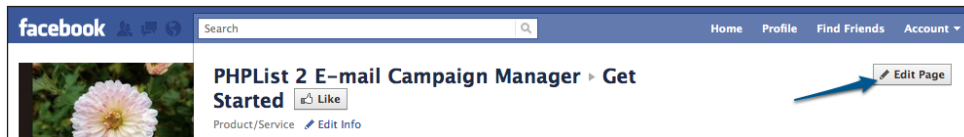
- Promote with an Ad
- View Insights
- Suggest to Friends
- Add to My Page**
- Add to My Page's Favorites
- Block App
- Share

When prompted, choose which of your pages you want to add the application to. In this example, we've created a page for this book, so we add the application to this page:

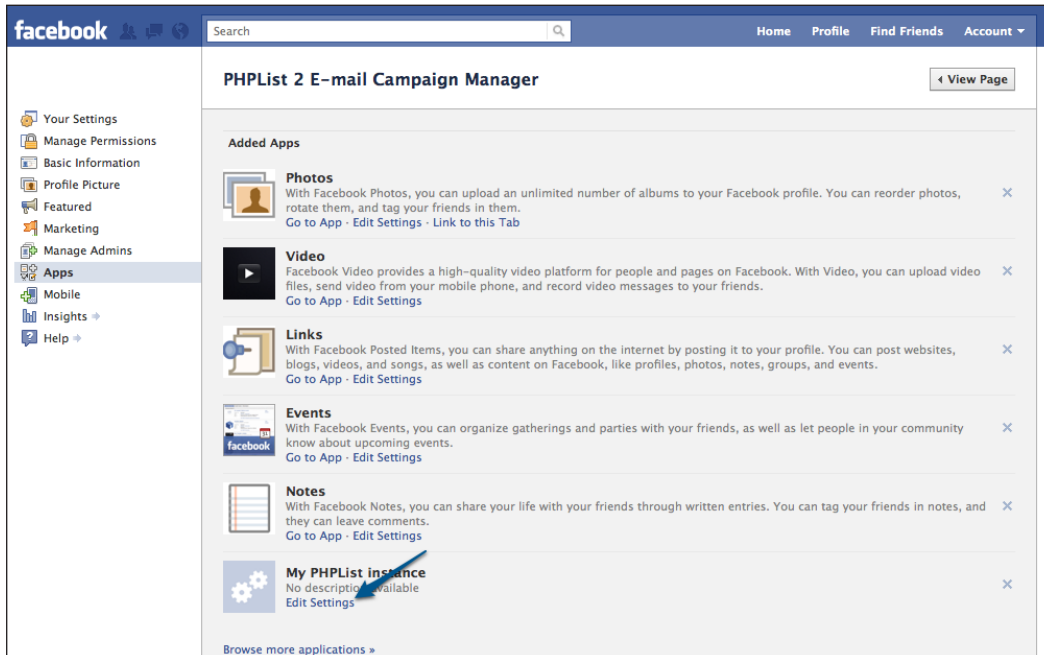


## Configuring the Facebook page

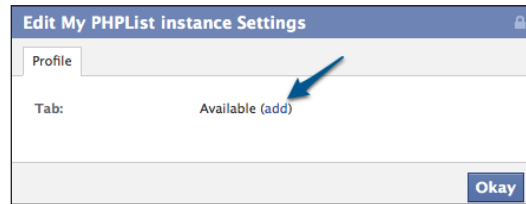
Now that we've added the application to our Facebook page, view the page and click on the **Edit Page** button:



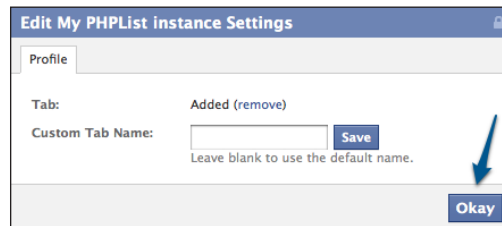
Under the **Apps** section, you'll find your application listed. Click on **Edit Settings**:



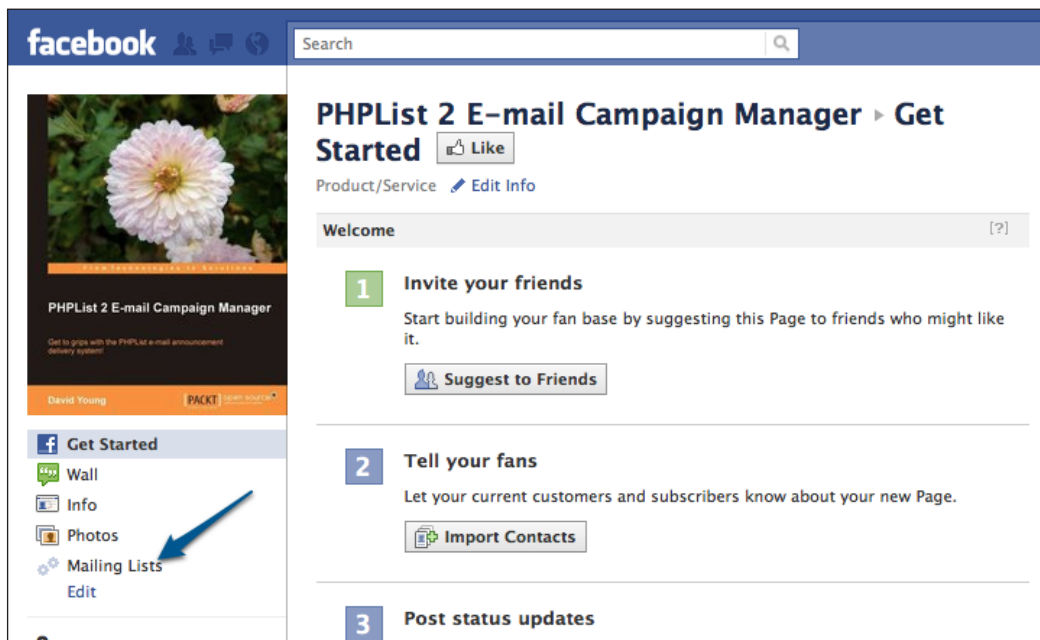
The only setting available to us is adding a tab, so click on **Add** to do so:



You are prompted for a custom name for your tab. The default name you chose when creating the application will be used, unless you enter a custom name. Click on **Okay** to complete:

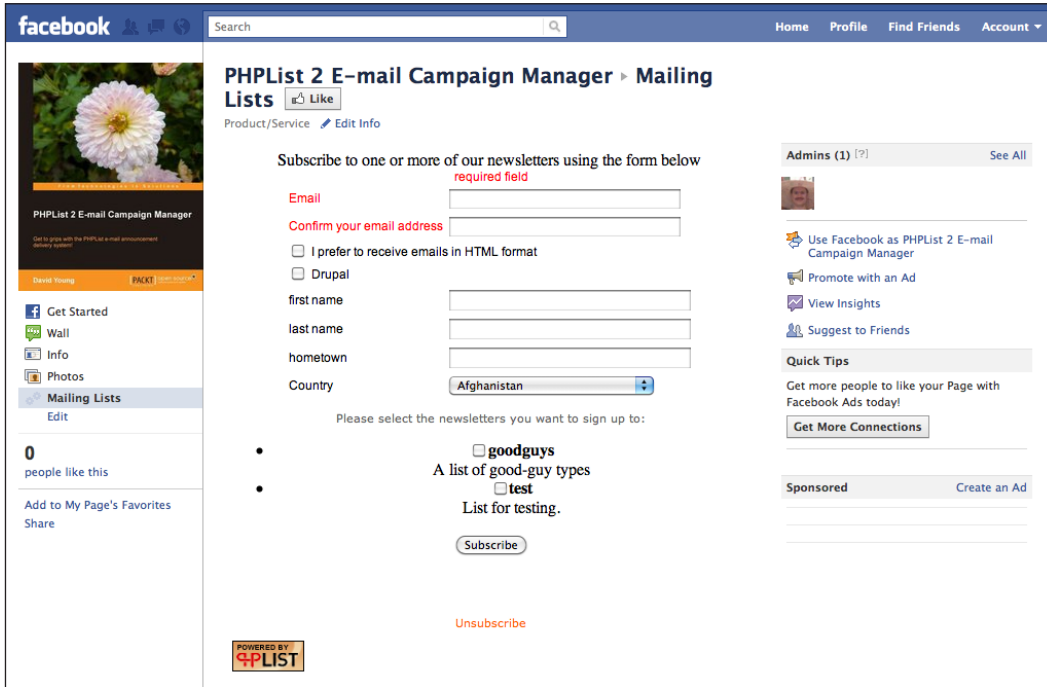


The next time you view your page, you'll notice an additional tab created for your application:





Clicking the tab will show you your phpList installation inside an iframe:

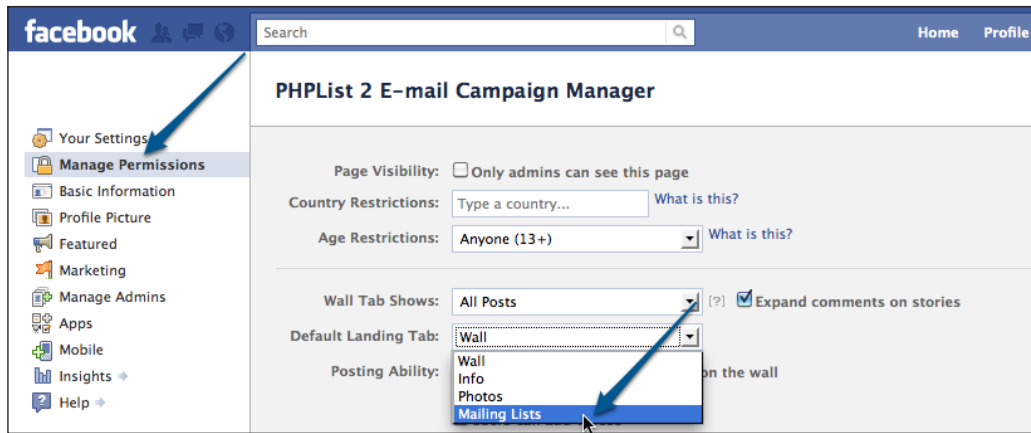


Obviously, any adjustments you make to the styling of the header and footer inside phpList will alter the way the phpList content is displayed on the Facebook page.

## Making phpList the default tab on your Facebook page

You may want the phpList page to be the default tab on your page, so that when someone visits the page, the first tab they see is your phpList sign up form.

To change your page's default tab, edit the page again and click on **Manage Permissions**. Set the **Default Landing Tab** to the tab created by your app and click on **Save**:



This tab will now be shown to visitors by default, as opposed to the **Wall** or **Info** tab.

## Summary

In this chapter, you've learned about integrating phpList with popular publishing platforms, including:

- Drupal
- WordPress
- Facebook

In the next chapter, we'll tackle managing users, lists, and administrators.



# 5

## List, User, and Administrator Management

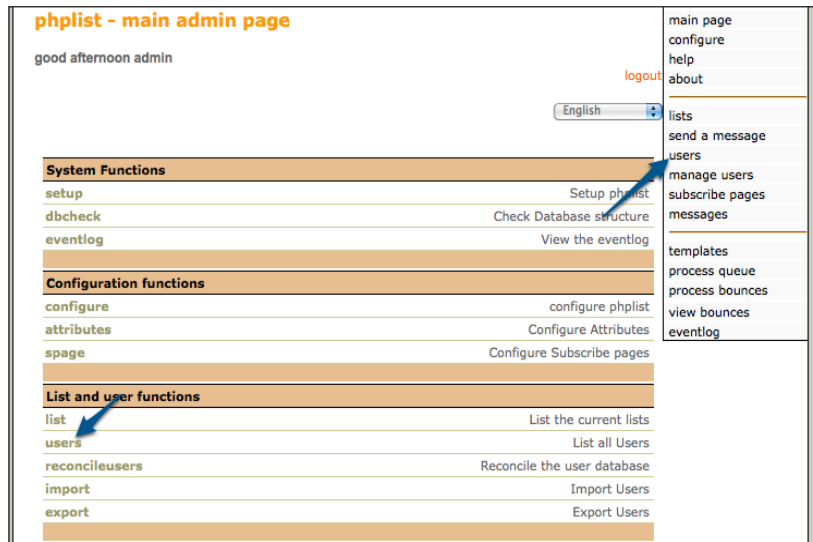
In the previous chapters, we've established our mailing list system and made it possible for our members to subscribe to. While phpList automatically takes care of many of the routine and repeatable parts of mailing list management, such as subscription, unsubscription, bounce processing, and so on, it remains the administrator's role to manage users, create lists, and possibly delegate some tasks to other administrators.

In this chapter, you'll learn to:

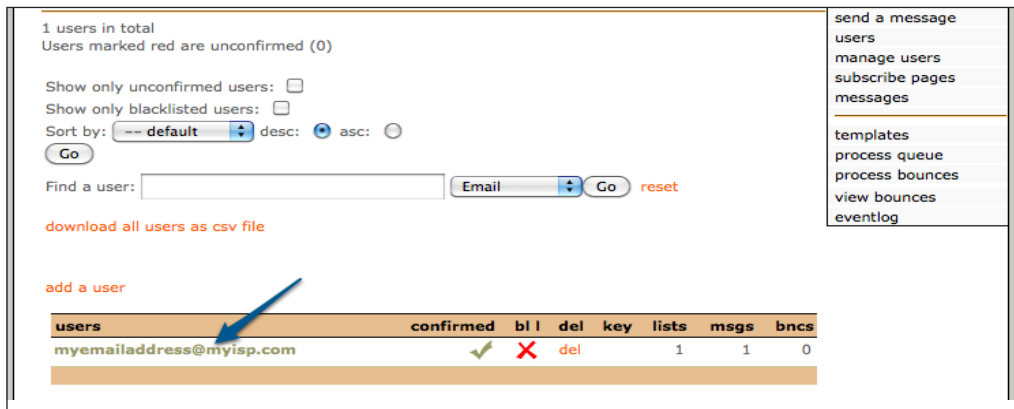
- Manage users either individually or in bulk
- Manage lists and their associated users
- Manage administrators and permission levels

# Individual user management

Let's start with creating some users. Click on the **users** link in the right-hand navigation panel or under the **List and user functions** heading to go to the **user management** page:



You are presented with the list of users (if there are too many users to display, phpList will prompt you to search for a subset of users). We have already added a user in the first chapter, so let's edit that user. Click on the user's e-mail address to edit their account:



## Updating a user's details and list memberships (as the user)

phpList provides a unique URL for each user to update their details, including their e-mail address, list memberships, and any other attributes you've defined. This URL can be added to the footer of your messages using the variable substitutions discussed in the following chapter.

To see what the user would see on this page, click on the **update page** link:

The screenshot shows a user management interface. At the top left, there is a link "back to the list of users". Below it, a warning message states "Delete will delete user and all listmemberships" with a "delete" link. The user's email address is "myemailaddress@myisp.com", and there are links for "update page", "unsubscribe page", and "history". The "User Details" section contains a table with the following data:

|                              |                          |
|------------------------------|--------------------------|
| ID                           | 1                        |
| Email                        | myemailaddress@myisp.com |
| Is this user confirmed (1/0) | 1                        |

On the right side, there is a vertical menu with the following items: "send a message", "users", "manage users", "subscribe pages", "messages", "templates", "process queue", "process bounces", "view bounces", and "eventlog".

You are redirected to a URL specific to this user's preferences. (This is the same URL the user would be directed to if they were to click their preferences link in an e-mail, to modify their subscription or formatting details):

The screenshot shows the user preferences page in a browser. The address bar contains the URL: `http://phplistpackt.com/lists/?p=preferences&uid=873958bedd85ed53f93f3dc672aea25e`. The page header features the "PHPLIST" logo and "powered by: [PHP] + [n]". A "PHPLIST" button is visible in the top right corner. The main content area contains the following form:

Please confirm that the information below is correct and click the "Update" button  
 required field

Email

Confirm your email address

I prefer to receive emails in HTML format

Country

Please select the newsletters you want to sign up to:

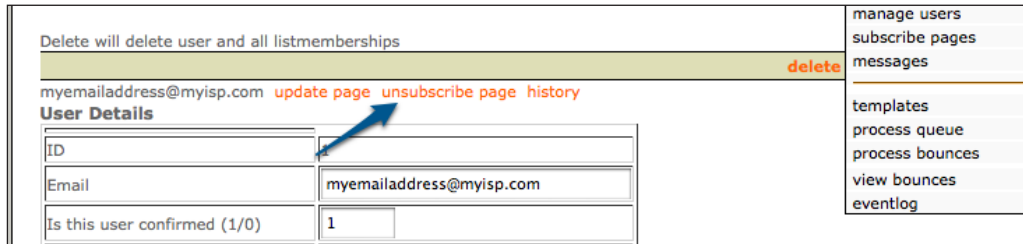
- test  
List for testing.
- test2

Unsubscribe

POWERED BY PHPLIST

## Unsubscribing a user (as the user)

Use your browser's back button to return to the **User Details** administration page and click the **unsubscribe** page link:



Delete will delete user and all listmemberships delete

myemailaddress@myisp.com [update page](#) [unsubscribe page](#) [history](#)

**User Details**

|                              |                          |
|------------------------------|--------------------------|
| ID                           |                          |
| Email                        | myemailaddress@myisp.com |
| Is this user confirmed (1/0) | 1                        |

- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

Like the update page, this is a unique URL that this user can visit to unsubscribe from all lists:



http://phplistpackt.com/lists/?p=unsubscribe&uid=873958bedd85ed53f93f3dc672aea25e

powered by: [PHP] + [mySQL]

PHPlist

**Unsubscribe from our newsletters**

Please enter a valid email address:

We are sorry to find you are not any longer interested in our newsletter. To help us improve our services, we would be grateful if you could tell us why:

**Please note:** this page is to stop receiving emails from our newsletter system **forever**. If you want to change your email, or if you want to change which newsletters you receive, please [update your preferences](#) instead

POWERED BY PHPLIST

## Examining a user's history (as administrator)

Use the **history** link to examine the user's history:

|   |                          |                 |
|---|--------------------------|-----------------|
| Delete will delete user and all listmemberships   |                          | manage users    |
| <a href="#">delete</a>  |                          | subscribe pages |
| myemailaddress@myisp.com <a href="#">update page</a> <a href="#">unsubscribe page</a> <a href="#">history</a> |                          | messages        |
| <b>User Details</b>   |                          | templates       |
| ID  | 1                        | process queue   |
| Email   | myemailaddress@myisp.com | process bounces |
| Is this user confirmed (1/0)  | 1                        | view bounces    |
|   |                          | eventlog        |

The user's history page shows every event associated with this user including their original subscription, every e-mail they've received (you can click the message ID to view it), as well as bounces, unsubscriptions, and so on:

|   |   |                     |  |                 |
|---|---|---------------------|--|-----------------|
| <b>phplist - history of a user</b>                |   |                     |  | main page       |
| good afternoon admin                              |   |                     |  | configure       |
| <a href="#">Click to examine this message</a>     |   |                     |  | help            |
| <a href="#">logout</a>                            |   |                     |  | about           |
| User <b>myemailaddress@myisp.com</b>              |   |                     |  | lists           |
| 1 messages sent to this user                      |   |                     |  | send a message  |
| <b>Messages</b>                                   | <b>sent</b>   | <b>bounce</b>       |  | users           |
| 1   | 20 Mar 2011 13:37:46  |                     |  | manage users    |
| <b>User Subscription History</b>                  |   |                     |  | subscribe pages |
| <b>Subscription History</b>                       | <b>ip</b>   | <b>date</b>         | <b>summary</b>   | messages        |
| 1   | 202.170.164.18  | 2011-03-20 13:14:36 | Update by admin<br>confirmed = 1<br>changed from 0<br>Country = Antarctica<br>changed from (no data) | templates       |
| <b>detail</b>                                     |   |                     |  | process queue   |
| List subscriptions:<br>Is now subscribed to: test |   |                     |  | process bounces |
| <b>info</b>                                       | HTTP_USER_AGENT = Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.15) Gecko/20110303 Firefox/3.6.15 |                     |  | view bounces    |
|   | HTTP_REFERER = http://phplistpackt.com/lists/admin/?page=user   |                     |  | eventlog        |


## Updating a user's details (as admin)

Of course, as the administrator, you don't need to use the update or unsubscribe link to manage the user – you can perform individual user operations from the user detail page. From this page, you can:

1. Confirm/unconfirm a user (users are only confirmed when they double option and only confirmed users receive e-mail).
2. Disable a user.



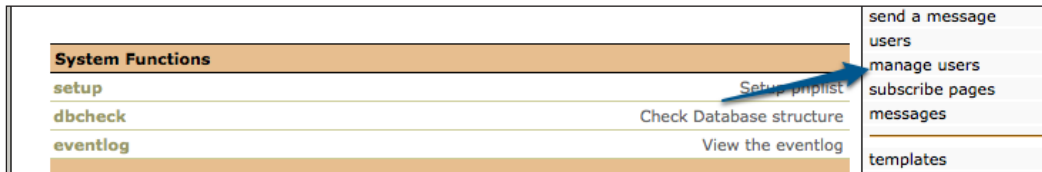
3. Alter the user's RSS frequency setting (we'll cover this in *Chapter 9, Advanced Features*).
4. Alter the user's HTML / plain text preference.
5. Alter any defined attributes.

 For all of the true/false values (that is, confirmation status or HTML preference), phpList expects you to enter 0 for false and 1 for true.

## Bulk user management

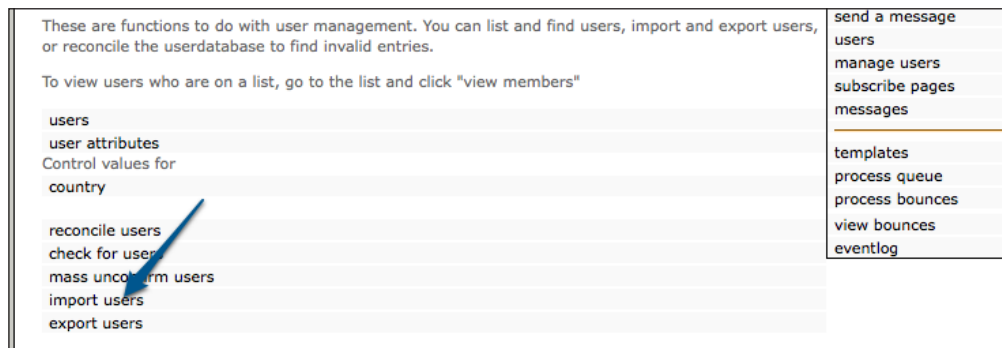
We've covered managing an individual user. phpList can also perform bulk actions on users, such as importing / exporting, moving between lists, and reconciling.

For bulk user management, click on the **manage users** link in the right-hand navigation panel:



## Importing users

Click on the **import users** link to display the options for importing users:



## Importing a simple list

The most common means of importing users is the uploading of a plain text file containing a list of e-mail addresses.

Click on the **import emails with the same values for attributes** link:

|  |   |
|--|---|
| <p><b>Import emails into existing lists</b><br/>There are four ways to import existing information:</p> <ul style="list-style-type: none"> <li>• <b>import emails with different values for attributes.</b> The list of emails can have attributes not already defined. They will be created automatically as "textline" attributes. You should use this option, if you are importing a spreadsheet/CSV file that has the attributes for the users in the columns and one user per line.</li> <li>• <b>import emails with the same values for attributes.</b> The list of emails will have to comply with the structure you have already set up in phplist. You should use this option if you are importing a simple list of emails. You can then specify the values for the attributes for each entry. They will be the same for everyone you are importing.</li> <li>• <b>import emails from an imap account.</b> This will search emails in your IMAP folders and add them. Only the Name of the person can be found as an attribute.</li> <li>• <b>import emails from another database.</b></li> </ul> | <ul style="list-style-type: none"> <li>send a message</li> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> <li>templates</li> <li>process queue</li> <li>process bounces</li> <li>view bounces</li> <li>eventlog</li> </ul> |
|--|---|

Let's assume that we want to import a list of customer's e-mail addresses from a text file named `customers_simple.txt` containing the following:

```
clarkkent@packtphplist.com
loislane@packtphplist.com
jimmyolsen@packtphplist.com
brucewayne@packtphplist.com
peterparker@packtphplist.com
lanalang@packtphplist.com
brucebanner@packtphplist.com
```

We'll enter the following options on the import page:

1. Test import first (make sure our file syntax is correct).
2. Send all subscribers a notification giving them the opportunity to confirm their subscription (unconfirmed subscribers will not receive messages).

3. Set all subscribers to HTML e-mail with a standard **Country** attribute

The screenshot shows the 'phplist - import emails' interface. The user is logged in as 'good morning admin'. The form includes a 'File containing emails:' field with the value '/code/customers\_simple.txt' and a 'Browse...' button. Below this are fields for 'Field Delimiter:' (default is TAB) and 'Record Delimiter:' (default is line break). A 'Test output:' checkbox is checked. The 'Send Notification email' section has 'I prefer to receive emails in HTML format' checked and 'Country' set to 'United States'. An 'import' button is at the bottom left. Annotations include a box around 'Adding users to list test', a note 'if you have multiple lists, this will be a checkbox per list' pointing to the 'Test output' checkbox, and an arrow pointing to the 'import' button.

The test process shows that the e-mails will be imported one-per-line as we expect. Click on the **back** link to return to the import page:

This time leave **Test output** unchecked, re-select your file and subscription options, and click on **import**. Instead of seeing each e-mail address listed on the success page (there could be tens of thousands after all), you're simply shown a total of the successful imports:

## Importing a complex list

You might have a more complex list that includes customer names and surnames, as well as their home town. Let's assume we want to import a list of customer details we've exported from another database, which includes their e-mail addresses, first and last name, and home town. We have the file `customers_complex.csv` in the `.CSV` (comma-separated values) format:

```
email,first name, last name, hometown
clarkkent@packtphplist.com,Clark,Kent, Krypton
loislane@packtphplist.com,Lois,Lane,Metropolis
jimmyolsen@packtphplist.com,Jimmy,Olsen,Metropolis
brucewayne@packtphplist.com,Bruce,Wayne,Gotham City
```

```
peterparker@packtphplist.com, Peter, Parker, New York  
lanalang@packtphplist.com, Lana, Lang, Smallville  
brucebanner@packtphplist.com, Bruce, Banner, Dayton
```

Note the column names are defined on the first line and that the e-mail column has to be named `email`.

phpList can import complex lists using the **import emails with the different values for attributes** link:

|   |   |
|---|---|
| <p><b>Import emails into existing lists</b><br/>There are four ways to import existing information:</p> <ul style="list-style-type: none"><li>• <b>import emails with different values for attributes.</b> The list of emails can have attributes not already defined. They will be created automatically as "textline" attributes. You should use this option, if you are importing a spreadsheet/CSV file that has the attributes for the users in the columns and one user per line.</li><li>• <b>import emails with the same values for attributes.</b> The list of emails will have to comply with the structure you have already set up in phplist. You should use this option if you are importing a simple list of emails. You can then specify the values for the attributes for each entry. They will be the same for everyone you are importing.</li><li>• <b>import emails from an imap account.</b> This will search emails in your IMAP folders and add them. Only the Name of the person can be found as an attribute.</li><li>• <b>import emails from another database.</b></li></ul> | <ul style="list-style-type: none"><li>send a message</li><li>users</li><li>manage users</li><li>subscribe pages</li><li>messages</li><hr/><li>templates</li><li>process queue</li><li>process bounces</li><li>view bounces</li><li>eventlog</li></ul> |
|---|---|

There are more options to choose from when importing e-mails with different values for attributes. We now have options regarding what to do with conflicts, duplicates, or invalid addresses. Fortunately, the defaults will do fine for our import. Note that we haven't created the first name, last name, or hometown attributes, but as they will be one of the fields we import, phpList will automatically create them as text fields.

Remember to specify your field separator—in this example, we are using the **CSV** format, so a comma separates our values. Run an input with **Test output** checked first to confirm that our syntax and options are correct:

English

Here you can add multiple emails to your lists. Attributes need to be identified on the first line in the file. They will be created as textline attributes if they don't exist.

[reset import session](#)

**Adding users to list test**

The file you upload will need to have the attributes of the records on the first line. Make sure that the email column is called "email" and not something like "e-mail" or "Email Address". Case is not important.

If you have a column called "Foreign Key", this will be used for synchronisation between an external database and the phplist database. The foreignkey will take precedence when matching an existing user. This will slow down the import process. If you use this, it is allowed to have records without email, but an "Invalid Email" will be created instead. You can then do a search on "invalid email" to find those records. Maximum size of a foreign key is 100.

**Warning:** the file needs to be plain text. Do not upload binary files like a Word Document.

|   |   |
|---|---|
| File containing emails:   | <input type="text" value="xde/customers_complex.csv"/> <input type="button" value="Browse..."/> |
| <small>The following limits are set by your server:<br/> Maximum size of a total data sent to server: <b>8M</b><br/> Maximum size of each individual file: <b>10M</b><br/> phplist will not process files larger than 1MB</small> |   |
| Field Delimiter:  | , (default is TAB)  |
| Record Delimiter:   | (default is line break)   |

If you check "Test Output", you will get the list of parsed emails on screen, and the database will not be filled with the information. This is useful to find out whether the format of your file is correct. It will only show the first 50 records.

Test output:

- lists
- send a message
- users
- manage users
- subscribe pages
- messages

---

- templates
- process queue
- process bounces
- view bounces
- eventlog

After clicking on the **Import** button, because there are no matching, pre-existing attributes, phpList will prompt you to create a new attribute for each column which it's imported, or to select from the list of existing attributes. Choose **Create new one** and click on **Continue**:

Here you can add multiple emails to your lists. Attributes need to be identified on the first line in the file. They will be created as textline attributes if they don't exist.

[reset import session](#)

Reading emails from file ... ..ok, 8 lines

Please identify the target of the following unknown columns

| Import Attributes | select            |
|-------------------|-------------------|
| first name        | -- Create new one |
| last name         | -- Create new one |
| hometown          | -- Create new one |

**CONTINUE**

- send a message
- users
- manage users
- subscribe pages
- messages

---

- templates
- process queue
- process bounces
- view bounces
- eventlog

phpList will show you which attributes have been identified, and which columns from your file should be mapped to these attributes. Once you're happy with the mapping, click on **confirm import** to continue:

English

Here you can add multiple emails to your lists. Attributes need to be identified on the first line in the file. They will be created as textline attributes if they don't exist.

[reset import session](#)

Reading emails from file ... ..ok, 8 lines

| Summary    | maps to    |
|------------|------------|
| email      | email      |
| first name | first name |
| last name  | last name  |
| hometown   | hometown   |

[close](#)

**7 lines will be imported**

[confirm import](#)

**Test Output**

```
email -> clarkkent@packtphplist.com
first name -> Clark
last name -> Kent
hometown -> Krypton

email -> loislane@packtphplist.com
first name -> Lois
last name -> Lane
hometown -> Metropolis

email -> jimmyolsen@packtphplist.com
first name -> Jimmy
last name -> Olsen
hometown -> Metropolis
```

In this case, as these e-mail addresses already existed in the database from the simple import, phpList updated the user records with the new attribute data that was imported:

Reading emails from file ... ..ok, 8 lines

All the emails already exist in the database and are member of the lists  
User data was updated for 7 users  
0 users were matched by foreign key, 7 by email

[import some more emails](#)

Examining one of the imported users would determine that we've successfully imported not only the e-mail address, but also the additional values, which can later be used to personalize their e-mails:

[back to the list of users](#)

---

Delete will delete user and all listmemberships [delete](#)

[lanalang@packtphplist.com](#) [update page](#) [unsubscribe page](#) [history](#)

**User Details**

|                                  |  |
|----------------------------------|--|
| ID                               | 21   |
| Email                            | lanalang@packtphplist.com                    |
| Is this user confirmed (1/0)     | <input type="checkbox" value="1"/>           |
| Is this user blacklisted         | <input type="checkbox" value="0"/>           |
| Number of bounces on this user   | <input type="text" value="0"/>               |
| Entered                          | 2011-04-04 21:02:57                          |
| Last Modified                    | 2011-04-04 21:02:57                          |
| Unique ID for User               | 20541fe0d3605e9b3845f088010bcd2e             |
| Send this user HTML emails       | <input type="checkbox" value="0"/>           |
| Which page was used to subscribe | <input type="text"/>                         |
| RSS Frequency                    | <input type="text"/>                         |
| Password                         | <input type="text"/>                         |
| Last time password was changed   | <input type="text"/>                         |
| Is this account disabled?        | <input type="checkbox" value="0"/>           |
| Additional data                  | <input type="text"/>                         |
| Foreign Key                      | <input type="text"/>                         |
| Country                          | -- no value <input type="button" value="v"/> |
| hometown                         | <input type="text" value="Smallville"/>      |
| last name                        | <input type="text" value="Lang"/>            |
| first name                       | <input type="text" value="Lana"/>            |

**Mailinglist Membership:**

test

[send a message](#)

[users](#)

[manage users](#)

[subscribe pages](#)

[messages](#)

---

[templates](#)

[process queue](#)

[process bounces](#)

[view bounces](#)

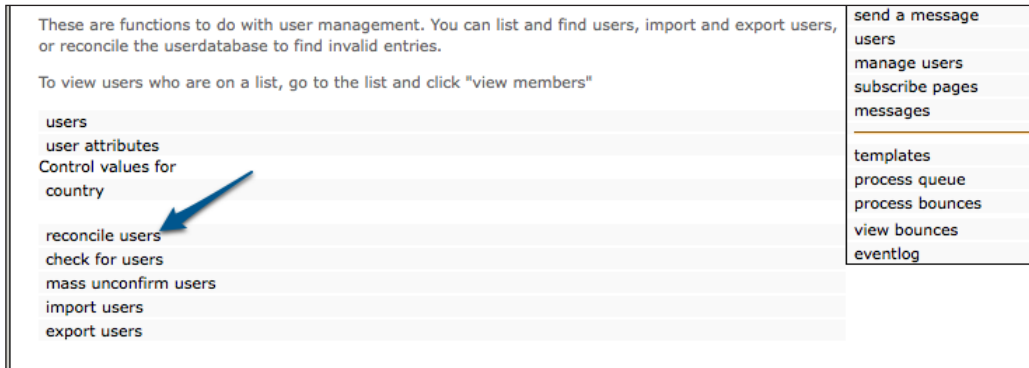
[eventlog](#)

## Reconciling users

When managing large numbers of users, it can be very helpful to run batch operations to perform bulk-updates on all (or a subset of) users.



To use the user reconciliation tools, click on the **reconcile users** link in the **manage users** page:



Many of the functions available are intended to address minor import issues. Here's a brief breakdown of what's possible:

1. **Delete all users who are not subscribed to any list:**

Either in testing or importing, you may have accidentally imported users, but not assigned them to any lists. This option will get rid of any of these "orphaned" users.

2. **Find users who have an invalid email:**

phpList has a function that checks whether an e-mail address is valid (that is, it can't contain spaces and has to contain an @ sign). This option will identify all users whose e-mail address is invalid.

3. **Make sure that all users have a unique ID:**

Under certain circumstances, users can be imported with the same unique ID. This option identifies these users and creates unique IDs for them

4. **Mark all users with an invalid email as unconfirmed:**

Instead of having phpList warn you every time it tries to send e-mails to invalid addresses, you may prefer to have these address marked as unconfirmed, so that they will not receive any further e-mails.

5. **Delete users who have an invalid email:**

If you choose not to bother with the invalid users, you could delete them altogether, keeping your user database free of clutter.

6. **Mark all users to receive HTML:**

If you don't want to offer your users the option to choose between HTML and text e-mails, you could bulk-change all users to receive HTML e-mails.

7. **Mark all users to receive text:**

If you don't want to offer your users the option to choose between HTML and text e-mails, you could bulk-change all users to receive text e-mails.

8. **Mark all users as confirmed:**

Perhaps you accidentally imported 10,000 e-mails as unconfirmed. Use this option to bulk-confirm all your users.

9. **To try to (automatically) fix emails for users who have an invalid email:**

phpList can try to automatically fix invalid e-mails by removing spaces and fixing common errors.

10. **Remove stale entries from the database:**

In addition to removing invalid users, this will also clean up some of the database tables.

11. **Merge duplicate users:**

You may have imported a user twice or once with a valid e-mail and once with an invalid e-mail that you've later fixed. This option will try to merge any duplicate users.

You also have the option at the bottom of the page to either delete or send a subscription reminder to any unconfirmed users who signed up during a particular date range.

## Exporting users

phpList's options for exporting users are quite straightforward. Either on the user management page (all users) or from the member's page of a particular list, there is a link to download users as a `csv` file.

## All users

Using the list of all users, you can globally export all your users details, across all lists

The screenshot shows the 'phpList - list of all users' interface. At the top, it says 'good morning admin' and 'logout'. There's a language dropdown set to 'English'. Below that, it states '15 users in total' and 'Users marked red are unconfirmed (5)'. There are checkboxes for 'Show only unconfirmed users' and 'Show only blacklisted users'. A 'Sort by' dropdown is set to 'default', with radio buttons for 'desc' (selected) and 'asc'. A 'Go' button is next to it. Below that is a search box with 'Email' dropdown and 'Go' and 'reset' buttons. A red link 'download all users as csv file' is highlighted with a blue arrow. At the bottom, there's an 'add a user' link and a table of users.

| users                        | confirmed | bl | del | key | lists | msgs | bncs |
|------------------------------|-----------|----|-----|-----|-------|------|------|
| myemailaddress@myisp.com     | ✓         | ✗  | del |     | 1     | 1    | 0    |
| brucebanner@packtphplist.com | ✓         | ✗  | del |     | 1     | 2    | 0    |
| peterparker@packtphplist.com | ✓         | ✗  | del |     | 1     | 2    | 0    |

## Specific list

Using a specific list's members page, you can export only the members of that list

### phplist - list membership

good morning admin

logout

English

Delete will remove this person from this list only

#### Members of goodguys

[back to this list](#)  
[download users on this list as a csv file](#)

Add a user:

12 Users on this list

Tag all users in this page

| Members  | confirmed | tag                      | del | # msgs |
|--|-----------|--------------------------|-----|--------|
| <a href="mailto:brucebanner@packtphplist.com">brucebanner@packtphplist.com</a> | ✓         | <input type="checkbox"/> | del | 2      |
| <a href="mailto:brucewayne@packtphplist.com">brucewayne@packtphplist.com</a>   | ✓         | <input type="checkbox"/> | del | 2      |
| <a href="mailto:clarkkent@packtphplist.com">clarkkent@packtphplist.com</a>     | ✓         | <input type="checkbox"/> | del | 2      |
| <a href="mailto:jimmyolsen@packtphplist.com">jimmyolsen@packtphplist.com</a>   | ✓         | <input type="checkbox"/> | del | 2      |

- main page
- configure
- help
- about

---

- lists
- send a message
- users
- manage users
- subscribe pages
- messages

---

- templates
- process queue
- process bounces
- view bounces
- eventlog

After clicking on the **download** link, you're prompted to select a date range and optionally choose which attributes will be exported. All the standard attributes (Email, HTML preference, and so on) will be optional as well as any custom attributes defined:

### phplist - export users

good morning admin

logout

English

You can export users based on date.  
If you click the "Export" button, you will get a popup box to download a file. This will be a TAB delimited text file containing the results of your export query.  
You can use this file in most spreadsheet applications.

#### Export users on goodguys

Date From:

Date To:

What date needs to be used:

When they signed up

When the record was changed

When they subscribed to

Select the columns to include in the export

ID

Email

- main page
- configure
- help
- about


---

- lists
- send a message
- users
- manage users
- subscribe pages
- messages

---

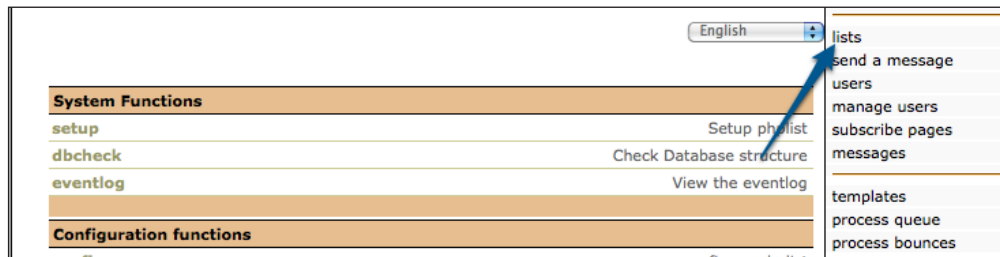
- templates
- process queue
- process bounces
- view bounces
- eventlog

Once you've clicked on **Download**, a text file will be downloaded which can be imported into a spreadsheet application.

 Note that while phpList's download links refer to downloading a CSV (comma-separated values), file the fields in the file will actually be delimited with tabs as opposed to commas.

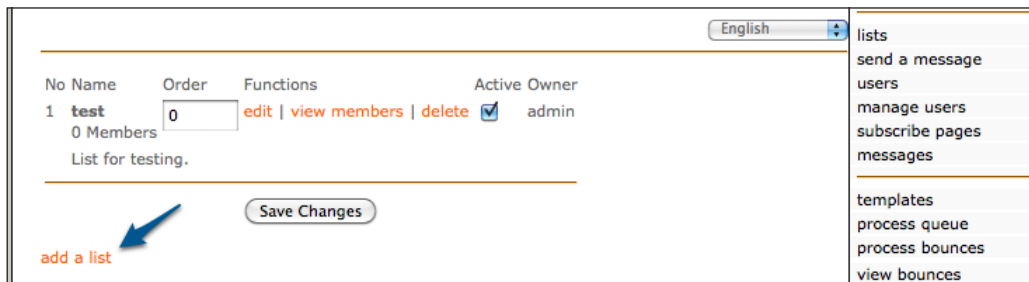
## Managing lists

To manage lists, use the **lists** link in the right-hand navigation panel:



## Creating a list

On the list page, all current active and inactive lists are displayed. In the introductory chapter, we activated the default "test" list. Let's now go into more detail and create a new list. Click on the **add a list** link to add a new list:



Enter a name for your new list, check the checkbox to make it active, and enter an optional description. If you had a huge selection of lists, you could manually arrange the order in which they are displayed by changing the value of the order. Click on **Save** to create your list:

|   |  |
|---|--|
| List name: <input type="text" value="goodguys"/>  | <ul style="list-style-type: none"> <li>send a message</li> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> </ul> <hr/> <ul style="list-style-type: none"> <li>templates</li> <li>process queue</li> <li>process bounces</li> <li>view bounces</li> <li>eventlog</li> </ul> |
| Check this box to make this list active (listed) <input checked="" type="checkbox"/>  |  |
| Order for listing <input type="text" value="0"/>  |  |
| Owner <input type="text" value="admin"/>  |  |
| List Description<br><div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <p>A list of good-guy types</p> </div> |  |
| <input type="button" value="Save"/> <input type="button" value="Reset"/>  |  |

You are returned to the **lists** page. We can tell that our list has been created, but that it currently has zero members:

| No | Name   | Order                          | Functions                    | Active                              | Owner |
|----|--|--------------------------------|------------------------------|-------------------------------------|-------|
| 1  | <b>test</b><br>8 Members<br>List for testing.            | <input type="text" value="0"/> | edit   view members   delete | <input checked="" type="checkbox"/> | admin |
| 3  | <b>goodguys</b><br>0 Members<br>A list of good-guy types | <input type="text" value="0"/> | edit   view members   delete | <input checked="" type="checkbox"/> | admin |

Now that we've created the list, we can edit it (the same information we just entered), delete it, or perform member operations on it.

## Performing member operations on a list

When you click on **view members** next to a particular list on the lists page, you are shown all (or a subset of) the members of that list. From here, you can bulk move or copy either all, or only selected, users to a different list.

In this example, let's move all the users we imported from the default **test** list to the new **goodguys** list which we just created. Click on **view members** next to the test list, and then click on **Tag all users in this page** to tag every e-mail address displayed, and then uncheck the entry for our test user – **myemailaddress@myisp.com**, as we don't want to move this one. Choose to move all tagged users to the **goodguys** list and click on **do it**:

The screenshot shows the 'Members of test' interface. At the top, there are links for 'back to this list' and 'download users on this list as a csv file'. Below that is an 'Add a user:' input field with an 'Add' button. It indicates '8 Users on this list' and has a checked checkbox for 'Tag all users in this page'. A table lists the members with columns for 'confirmed', 'tag', 'del', and '# msgs'. The user 'myemailaddress@myisp.com' has its 'tag' checkbox unchecked. Below the table are sections for 'What to do with "Tagged" users:' and 'What to do with all users:'. In the 'Tagged' section, 'Move' is selected and set to 'goodguys'. In the 'all users' section, 'Nothing' is selected. A blue arrow points to the 'do it' button at the bottom.

| Members                      | confirmed | tag                                 | del | # msgs |
|------------------------------|-----------|-------------------------------------|-----|--------|
| brucebanner@packtphplist.com | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| brucewayne@packtphplist.com  | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| clarkkent@packtphplist.com   | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| jimmyolsen@packtphplist.com  | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| lanalang@packtphplist.com    | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| loislane@packtphplist.com    | ✓         | <input checked="" type="checkbox"/> | del | 0      |
| myemailaddress@myisp.com     | ✓         | <input type="checkbox"/>            | del | 1      |
| peterparker@packtphplist.com | ✓         | <input checked="" type="checkbox"/> | del | 0      |

After clicking on **do it**, we're presented with the results of the bulk action. Note that there is only one subscriber remaining in the **test** list:

| Processing ....<br>7 users were moved to goodguys<br>1 Users on this list<br><br><input type="checkbox"/> Tag all users in this page   | process queue<br>process bounces<br>view bounces<br>eventlog |                          |     |        |        |                          |   |                          |     |   |  |
|--|--|--------------------------|-----|--------|--------|--------------------------|---|--------------------------|-----|---|--|
| <table border="1"> <thead> <tr> <th>Members</th> <th>confirmed</th> <th>tag</th> <th>del</th> <th># msgs</th> </tr> </thead> <tbody> <tr> <td>myemailaddress@myisp.com</td> <td>✓</td> <td><input type="checkbox"/></td> <td>del</td> <td>1</td> </tr> </tbody> </table> | Members  | confirmed                | tag | del    | # msgs | myemailaddress@myisp.com | ✓ | <input type="checkbox"/> | del | 1 |  |
| Members  | confirmed  | tag                      | del | # msgs |        |                          |   |                          |     |   |  |
| myemailaddress@myisp.com   | ✓  | <input type="checkbox"/> | del | 1      |        |                          |   |                          |     |   |  |
| close ↗  |  |                          |     |        |        |                          |   |                          |     |   |  |

Reviewing the list management page will also indicate that the users have been successfully moved:

| No  | Name     | Order | Functions                    | Active                              | Owner |
|---|----------|-------|------------------------------|-------------------------------------|-------|
| 1   | test     | 0     | edit   view members   delete | <input checked="" type="checkbox"/> | admin |
| <div style="border: 1px solid blue; padding: 2px;">1 Members</div> List for testing.        |          |       |                              |                                     |       |
| 3   | goodguys | 0     | edit   view members   delete | <input checked="" type="checkbox"/> | admin |
| <div style="border: 1px solid blue; padding: 2px;">7 Members</div> A list of good-guy types |          |       |                              |                                     |       |

users  
 manage users  
 subscribe pages  
 messages  
  
 templates  
 process queue  
 process bounces  
 view bounces

## Managing administrators

If you have a few lists that you manage yourself, then you might be quite comfortable simply using the default super-admin account for all administration work. However, if your phpList installation encompasses a lot of lists, you may have delegated management of some of these lists to other (restricted) admins or even other super-admins.

## Creating a restricted admin

Let's create another administrator to manage our test2 list. On the main page, click on the **admins** link to go to the **list administrators** page:

| Administrator functions |   |
|-------------------------|---|
| <b>admins</b> ←         | Add, edit and remove Administrators     |
| adminattributes         | Configure attributes for administrators |

close ↗



On this page, you can see the current admin user listed. Click on **add new admin** to create another administrator:

The screenshot shows the 'phplist - list administrators' interface. At the top left, it says 'good morning admin'. Below that, there's a link 'add new admin' and '1 Administrators'. A search box 'Find an admin:' with a 'Go' button is present. A table lists administrators, with one entry 'admin' and a 'close' link. At the bottom, there are links for 'add new admin' (highlighted with a blue arrow) and 'import list of admins'. A sidebar on the right contains a menu with items like 'main page', 'configure', 'help', 'about', 'lists', 'send a message', 'users', 'manage users', 'subscribe pages', 'messages', 'templates', 'process queue', 'process bounces', 'view bounces', and 'eventlog'.

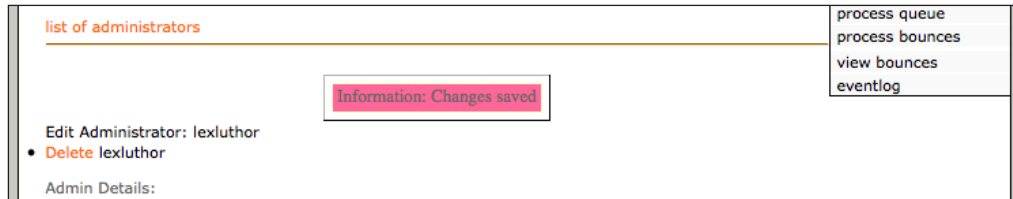
To start off with, we'll enter a few basic details about our new admin. Include their login name, e-mail address, and password. Unless you want this admin to be granted all possible rights, enter 0 (that is, no) for the **Super Admin** field, and likewise for the **disabled** field. Click on **Save Changes** to save this admin account:

The screenshot shows the 'Add a new Administrator' form. The form has the following fields and values:

|                                |                           |
|--------------------------------|---------------------------|
| ID                             |                           |
| Login Name (max 25 chars)      | lexluthor                 |
| Normalised loginname           |                           |
| Email                          | lexluthor@phlistpackt.com |
| Time Created                   |                           |
| Time modified                  |                           |
| Modified by                    |                           |
| Password                       | capesareforweenies        |
| Last time password was changed |                           |
| Is this user Super Admin?      | 0                         |
| Is this account disabled?      | 0                         |

A 'Save Changes' button is located at the bottom of the form. A blue arrow points to the 'Is this user Super Admin?' field.

You'll receive a confirmation message to indicate that the admin has been created and will be returned to the same page to verify the admin permissions:



## Setting admin permissions

There are fifty access levels that can be assigned to an administrator. phpList has a sensible default of allowing administrators free reign on their own lists, but limited access to the global administration options.

Permissions are assigned by choosing one of the following levels: None, All, View, and Owner, and associating them with the fifty different types of admin page access.

An overview:

- **None** – no access. An administrator won't even see links to pages he/she has no rights to and any attempts to access these pages by URL would fail.
- **All** – full access. If an admin can see an "all" page, he/she has full rights to do whatever that page controls. All rights to the system upgrade page would allow an administrator to perform phpList database upgrades.
- **View** – read-only access. This is currently limited to viewing user details, user lists, and list membership lists, and is of limited usefulness.

- **Owner** – full access to admin's own lists. This is the access level most commonly assigned to sub-admins. It allows them to perform all admin functions on the lists assigned to them, without access to manage other lists, or any other global options.

Set permissions to view pages in the system:

- **All:** admin has access to page without restrictions
- **View:** admin can view content of pages, but not change anything. This currently only workfor the "user", "users" and "members" pages.
- **None:** admin cannot see this page
- **Owner:** admin can see this page, but only see the content of the lists they own

Note: Admin Password must be at least 4 characters long

Access Details:

| Access Privileges |               |                                  |                       |                       |                                  |
|-------------------|---------------|----------------------------------|-----------------------|-----------------------|----------------------------------|
| Type              | Page          | none                             | all                   | view                  | owner                            |
| admin             | admin         | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| admin             | admins        | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>            |
| clickstats        | statsmgt      | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| clickstats        | mclicks       | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| clickstats        | uclicks       | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| clickstats        | userclicks    | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| clickstats        | mviews        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| clickstats        | statsoverview | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| list              | editlist      | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |

Adjust the permissions as required and click on **Save Changes** at the bottom of the list:

|      |               |                                  |                       |                       |                       |
|------|---------------|----------------------------------|-----------------------|-----------------------|-----------------------|
| user | export        | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| user | massunconfirm | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |



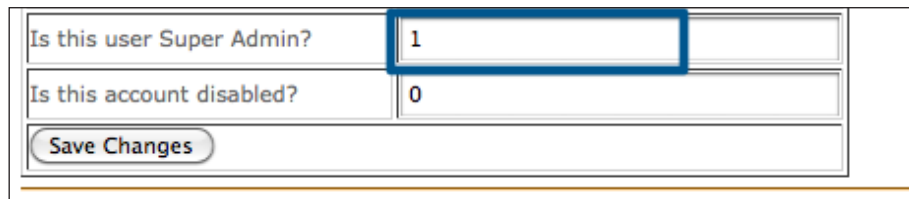
#### Setting default permissions

To change the default admin permissions, make your required adjustments and then click on **Set these permissions as default**. Any subsequently created admins will inherit these new defaults.

## Creating a super admin

A super admin is an administrator with full administrative rights and no restrictions applied. phpList's default admin account is a super-admin. Delegate this role carefully, because a super-admin can edit (or create) any other super-admin's account and change any configuration settings.

To create a super-admin, either add a new user or edit an existing user, and change the value of **Is this user Super Admin?** to 1:



The screenshot shows a form with two input fields. The first field is labeled "Is this user Super Admin?" and contains the value "1". The second field is labeled "Is this account disabled?" and contains the value "0". Below the fields is a "Save Changes" button.

|                           |   |
|---------------------------|---|
| Is this user Super Admin? | 1 |
| Is this account disabled? | 0 |

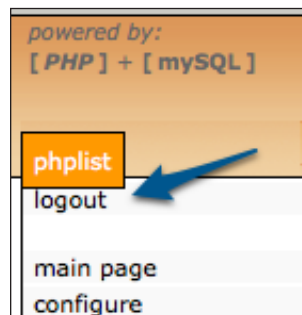
Save Changes

## Assigning administrators to lists

The final component of managing administrators is assigning admins to lists. Edit a list using the lists management page and change the owner using the drop-down menu:

## Testing a new administrator

To confirm that the new administrator has been created successfully, log out of phpList by clicking the **logout** link at the top of the navigation panel, and re-login as the user you just created:



When you login as the new admin, you'll see that there is only a subset of management options available, and only those that relate directly to this admin are shown:



## Summary

In this chapter, you've learned about managing users, lists, and admins, specifically:

- Managing users either individually or in bulk
- Managing lists and their associated users
- Managing administrators and permission levels

In the next chapter, we'll talk about personalizing messages to your subscribers based on their unique attributes.

# 6

## Personalizing E-mail Body

Now that we've covered creation of lists and attributes, let's look at how to use these lists and attributes to personalize the e-mail we send to our subscribers. Personalizing e-mail messages makes them more engaging to readers and increases the effectiveness of your newsletters.

In this chapter, you'll learn to:

- Enhance messages with built-in placeholders
- Personalize messages using member attributes
- Send messages to subsets of lists based on attributes
- Use message templates for consistent layout and styling

### Enhancing messages using built-in placeholders

For simple functionality's sake, we generally want our phpList messages to contain at least a small amount of customization. For example, even the default footer, which phpList attaches to messages, contains three placeholders, customizing each message for each recipient:

```
--  
If you do not want to receive any more newsletters, [UNSUBSCRIBE]  
  
To update your preferences and to unsubscribe, visit [PREFERENCES]  
  
Forward a Message to Someone [FORWARD]
```

The placeholders [UNSUBSCRIBE], [PREFERENCES], and [FORWARD] will be replaced with unique URLs per subscriber, allowing any subscriber to immediately unsubscribe, adjust their preferences, or forward a message to a friend simply by clicking on a link.

There's a complete list of available placeholders documented on phpList's wiki page at <http://docs.phplist.com/Placeholders>. Here are some of the most frequently used ones:

- [CONTENT]: Use this while creating standard message templates. You can design a styled template which is re-used for every mailing and the [CONTENT] placeholder will be replaced with the unique content for that particular message.
- [EMAIL]: This is replaced by the user's e-mail address. It can be very helpful in the footer of an e-mail, so that subscribers know which e-mail address they used to sign up for list subscription.
- [LISTS]: The lists to which a member is subscribed. Having this information attached to system confirmation messages makes it easy for subscribers to manage their own subscriptions. Note that this placeholder is only applicable in system messages and not in general list messages.
- [UNSUBSCRIBEURL]: Almost certainly, you'll want to include some sort of "click here to unsubscribe" link on your messages, either as a pre-requisite for sending bulk mail (perhaps imposed by your ISP) or to avoid users inadvertently reporting you for spamming.
- [UNSUBSCRIBE]: This placeholder generates the entire hyperlink for you (including the link text, "unsubscribe"), whereas the [UNSUBSCRIBEURL] placeholder simply generates the URL. You would use the URL only if you wanted to link an image to the unsubscription page, as opposed to a simple link, or if you wanted the HTML link text to be something other than "unsubscribe".
- [USERTRACK]: This inserts an invisible tracker image into HTML messages, helping you to measure the effectiveness of your newsletter (discussed in greater depth in the next chapter).

You might combine several of these placeholders to add a standard signature to your messages, as follows:

```
--  
You ([EMAIL]) are receiving this message because you subscribed to one  
or more of our mailing lists.  
  
We only send messages to subscribers who have requested and confirmed  
their subscription (double-opt-in). You can adjust your list  
membership at any time by clicking on [PREFERENCES] or unsubscribe  
altogether by clicking on [UNSUBSCRIBE].  
--
```

---

## Placeholders in confirmation messages

Some placeholders (such as [LISTS]) are only applicable in confirmation messages (that is, "thank you for subscribing to the following lists..."). These placeholders allow you to customize the following:

- Request to confirm: Sent initially to users when they subscribe, confirming their e-mail address and subscription request
- Confirmation of subscription: Sent to users to confirm that they've been successfully added to the requested lists (after they've confirmed their e-mail address)
- Confirmation of preferences update: Sent to users to confirm their updates when they change their list subscriptions/preferences themselves
- Confirmation of unsubscription: Sent to users after they've unsubscribed to confirm that their e-mail address will no longer receive messages from phpList

## Personalizing messages using member attributes

Apart from the built-in placeholders, you can also use any member attributes to further personalize your messages. Say you captured the following attributes from your new members:

- First Name
- Last Name
- Hometown
- Favorite Food

You could craft a personalized message as follows:

```
Dear [FIRST NAME],
```

```
Hello from your friends at the Funky Town Restaurant. We hope the  
[LAST NAME] family is well in the friendly town of [HOMETOWN].
```

```
If you're ever in the mood for a fresh [FAVORITE FOOD], please drop in  
- we'd be happy to have you!
```

```
...
```



This would appear to different subscribers as:

Dear Bart,

Hello from your friends at the Funky Town Restaurant. We hope the Simpson family is well in the friendly town of Springfield.

If you're ever in the mood for a fresh pizza, please drop in - we'd be happy to have you!

...

Or:

Dear Clark,

Hello from your friends at the Funky Town Restaurant. We hope the Kent family is well in the friendly town of Smallville.

If you're ever in the mood for a fresh Krypto-Burger, please drop in - we'd be happy to have you!

...

If a user doesn't have an attribute for a particular placeholder, it will be replaced with a blank space. For example, if user "Mary" hadn't entered any attributes, her message would look like:

Dear,

Hello from your friends at the Funky Town Restaurant. We hope the family is well in the friendly town of .

If you're ever in the mood for a fresh , please drop in - we'd be happy to have you!

...

If the attributes on your subscription form are optional, try to structure your content in such a way that a blank placeholder substitution won't ruin the text. For example, the following text will look awkward with blank substitutions:

Your name is [FIRST NAME], your favorite food is [FAVORITE FOOD], and your last name is [LAST NAME]

Whereas the following text would at least "degrade gracefully":

Your name: [FIRST NAME]  
Your favorite food: [FAVORITE FOOD]  
Your last name [LAST NAME]

## Sending messages to subsets of lists based on attributes

The techniques we've discussed so far apply to messages sent to all the members of a particular list. You can also use user attributes to define a target subset of members to which a particular message should be sent.

Let's say you wanted to e-mail only members who lived in Albania, whose favorite food is pizza. When composing your message, click on the **Criteria** tab to add criteria based on attributes. Select the applicable attributes and remember to check the **Use this one** checkboxes where applicable:

Content
Format
Scheduling
Criteria
Lists
Misc

templates  
 process queue  
 process bounces  
 view bounces  
 eventlog

**Select the criteria for this message:**

1. to use a criteria, check the box next to it
2. then check the radio button next to the attribute you want to use
3. then choose the values of the attributes you want to send the message to *Note: Messages will be sent to people who fit to **Criteria 1 AND Criteria 2** etc*

---

**Criterion 1** Use this one

Country is Afghanistan  
Albania  
Algeria  
American Samoa

Favorite Food is Spagetti  
Pizza  
Dry Toast  
Burgers

**Criterion 2** Use this one

Country is Afghanistan  
Albania  
Algeria  
American Samoa

Favorite Food is Spagetti  
Pizza  
Dry Toast  
Burgers

Send Test Message to email address(es): myemailaddress@myisp.com  
(comma separated addresses - all must be users)

Save Changes

After clicking on **Save Changes**, you can proceed to compose and send the message as you normally would. It will only be sent to users whose criteria match your selections.

Note that criteria-based selections don't work with plain text attributes, as the members' attributes can't be matched against previously known values. Only select, radio, or checkbox attributes are available.

## **Increasing the amount of criteria available**

By default, phpList only allows you two selection criteria when sending messages based on user attributes. To increase the amount of criteria available, you'll need to edit `config/config.php`.

On line #296 in `config/config.php`, change the 2 in the line below to the maximum amount of criteria you need:

```
define ("NUMCRITERIAS",2);
```

Let's assume you want a maximum of 5 criteria available for selection (you don't have to use them all). Change the `NUMCRITERIAS` variable as follows:

```
define ("NUMCRITERIAS",5);
```

Now while selecting criteria when composing a message, there are five criteria available for use:

**Select the criteria for this message:** eventlog

1. to use a criteria, check the box next to it
2. then check the radio button next to the attribute you want to use
3. then choose the values of the attributes you want to send the message to *Note: Messages will be sent to people who fit to **Criteria 1 AND Criteria 2** etc*

---

**Criterion 1** Use this one

Country is Afghanistan  
Albania  
Algeria  
American Samoa

Favorite Food is Spagetti  
Pizza  
Dry Toast  
Burgers

---

**Criterion 2** Use this one

Country is Afghanistan  
Albania  
Algeria  
American Samoa

Favorite Food is Spagetti  
Pizza  
Dry Toast  
Burgers

---

**Criterion 3** Use this one

Country is Afghanistan  
Albania  
Algeria  
American Samoa

Favorite Food is Spagetti  
Pizza  
Dry Toast  
Burgers

---

**Criterion 4** Use this one

Country is Afghanistan  
Albania

## Using message templates

So far, we've used placeholders and attributes to customize individual messages. phpList also allows us to create message templates, so that all the repetitive parts of a message (styling, footer, and so on) can remain consistent and "out of the way".

## Creating a message template

To create or manage message templates, click on the **templates** link on the right-hand navigation panel:

|                                |  |                          |  |
|--------------------------------|--|--------------------------|--|
| <b>System Functions</b>        |  |                          |  |
| setup                          |  | Setup phplist            |  |
| dbcheck                        |  | Check Database structure |  |
| eventlog                       |  | View the eventlog        |  |
|                                |  | close ↗                  |  |
| <b>Configuration functions</b> |  |                          |  |
| configure                      |  | configure phplist        |  |

users

manage users

subscribe pages

messages

**templates**

process queue

process bounces

view bounces

You are taken to the **phplist – templates in the system** page. All the templates you've defined will be listed here. Click on **add new template** to create a new template:

**phplist - templates in the system**

good evening admin

logout

English

Here you can define templates that can be used to send the emails to the lists. A template is an HTML page with somewhere the *Placeholder* **[CONTENT]**. This will be the place where the text for the email will be inserted.

Additionally to **[CONTENT]**, you can add **[FOOTER]** and **[SIGNATURE]** to insert the footer information and the signature of the message, but they are optional.

Images in your template will be included in the emails. If you add images to the content of your messages (when you send it), they need to include a complete URL, and will not be included in the email.

**User Tracking**

To facilitate user tracking, you can add **[USERID]** to your template which will be replaced by an identifier for a user. This will only work when sending the email as HTML. You will need to set up some URL that will receive the ID. Alternatively you can use the builtin usertracking of phplist. To do this add **[USERTRACK]** to your template and an invisible link will be added to the email to keep track of Views of the email.

**Images**

Any reference to an image that does not start with "http://" can (and should) be uploaded for inclusion in the email. It is advised to use only few images and make them very small. If you upload your template, you will be able to add your images. References to images to be included should be from the same directory, ie `` and not ``

No template have been defined

add new template

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

Either upload a pre-existing HTML file to use as a template or compose one using the WYSIWYG editor. Remember to insert the placeholder **[CONTENT]** where you want the dynamic content to be inserted when the message is sent:

list of templates

English

Title of this template: GoodGuys Inc

Content of the template: [CONTENT]

The content should at least have **[CONTENT]** somewhere. You can upload a template file or paste the text in the box below

Template file: Browse...

Style: Format: Normal Font: Size:

GoodGuys Inc.

[UNSUBSCRIBE], [PREFERENCES]

- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

Below the message compose window, above the **Save Changes** button, are options controlling the checking of external links and URLs:

|   |                          |
|---|--------------------------|
| Check that all links have a full URL        | <input type="checkbox"/> |
| Check that all images have a full URL       | <input type="checkbox"/> |
| Check that all external images exist        | <input type="checkbox"/> |
| <input type="button" value="Save Changes"/> |                          |

If you reference local images in your template, phpList will prompt you to upload these; in that case, they will be attached to each message upon sending.

Click on **Save Changes** to save your template. In this example, phpList acknowledges that there are no local images referenced in the template. Click on **send a message** to compose a new message using this template:

|  |                |
|--|----------------|
| English                                | lists          |
| Template saved                         | send a message |
| Template does not contain local images | users          |
|  | manage users   |

Compose your message (just the dynamic content), save at least once, and then click on the **Format** tab to select a template:

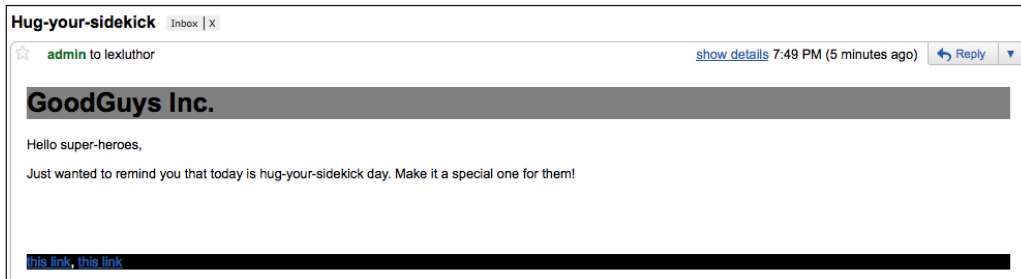
|  |                 |
|--|-----------------|
| <b>Content</b> <b>Format</b> <b>Scheduling</b> <b>Criteria</b> <b>Lists</b> <b>Misc</b>  | templates       |
| ? Subject: Hug-your-sidekick   | process queue   |
| ? From Line: admin myemailaddress@myisp.com  | process bounces |
| ? Message.   | view bounces    |
|  | eventlog        |
| Source   Paste   Undo   Redo   Bold   Italic   Underline   ABC   x <sub>2</sub>   x <sup>2</sup>   Bulleted List   Numbered List   Indent Left   Indent Right   Decrease Indent   Increase Indent   Text Color   Background Color   Link   Unlink   Image   Table   Table of Contents   Print   Help |                 |
| Style   Format Normal   Font   Size  |                 |
| Hello super-heroes,<br>Just wanted to remind you that today is hug-your-sidekick day. Make it a special one for them!  |                 |

Select your template and then continue to send the message as normal:



The screenshot shows a configuration interface with tabs for Content, Format, Scheduling, Criteria, Lists, and Misc. The 'Criteria' tab is active. Below the tabs, there are options for 'Send As: HTML' (radio button) and 'text' (radio button, selected). A blue arrow points to the 'Use Template:' dropdown menu, which is set to 'GoodGuys Inc'. Below this, there is a 'Send Test Message' button and an email address field containing 'myemailaddress@myisp.com'. A note below the field says '(comma separated addresses - all must be users)'. On the right side, there is a sidebar with links: templates, process queue, process bounces, view bounces, and eventlog.

The content you entered replaces the [CONTENT] placeholder in the template:



## Summary

In this chapter, you've learned to use built-in placeholders and user attributes to personalize the e-mails you send to your subscribers. You've learned about using built-in placeholders and custom attributes, selecting criteria when sending messages, and taking advantage of templates to provide a consistent look to your messages.

In this chapter, we have covered:

- Enhancing messages with built-in placeholders
- Personalizing messages using member attributes
- Sending messages to subsets of lists based on attributes
- Using message templates for consistent layout and styling

In the next chapter, we'll talk about measuring the effectiveness of your newsletter.

# 7

## Measuring Effectiveness of Newsletters

Effectiveness of a newsletter is commonly measured by the amount of delivered messages that are actually opened (and presumably read) or the percentage of readers who "click through" on a link in the message.

Your own definition of effectiveness will vary depending on your newsletter-publishing goals, but you will no doubt want to know as much as you can about which of your messages are opened, read, and which links are clicked.

In this chapter, you will learn how to implement and use the user-tracking and click-tracking features of phpList. These include:

- Enabling user tracking (we refer to user tracking as the visibility of whether or not a message has been opened by a particular recipient)
- Enabling click tracking (we refer to click tracking as the visibility of which links in either our text or HTML messages are clicked on, by which users)
- Examining user/click statistics in greater detail using the "statistics" menu



### **Privacy and transparency note**

While it's a common practice for newsletter publishers to gather as much information as they can from their readers, it's recommended that you explicitly declare what you will and won't record, in a privacy policy which is both published online and referred to in your message content.



## User tracking

User tracking can be used to tell whether a unique recipient of an HTML-formatted message has opened the message in their e-mail client. phpList tracks user activity by embedding a tiny (1 x 1 pixel) transparent image into each HTML-formatted e-mail that it sends. This image, instead of being an actual image file, is a unique URL to a phpList script, whose function when loaded is to record a "hit" in a database table against a unique message ID.

phpList can then examine that database table to tell which messages destined for which users have been opened.

The embedding tracking image will look something like this in the source of your HTML message:

```

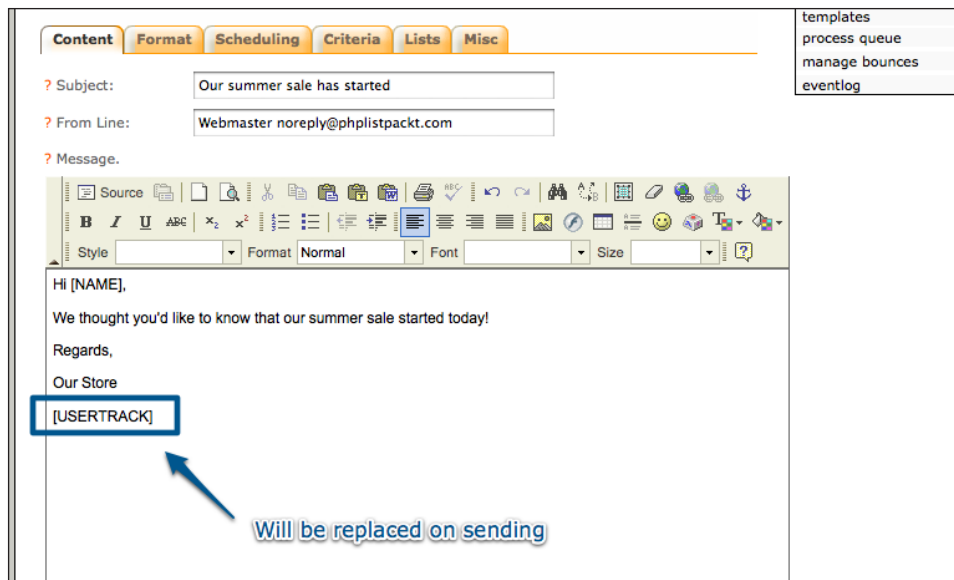
```

As this technique relies on the e-mail being displayed in HTML, it won't work on users who receive your messages in plain text format.

This common technique for e-mail opening tracking suffers from a few limitations – if the user reading the e-mail is (a) not connected to the Internet, or (b) filtering images in their e-mail client, the tracking image won't be fetched and phpList won't record an "open" result against this e-mail.

## Sending a message with user tracking enabled

To enable user tracking on a message, add the placeholder [USERTRACK] anywhere in the HTML body of the message. When the message is sent, phpList will replace this holder with a unique tracking "image" for this user.



#### Turning on user tracking by default

To turn on user tracking by default, you can add this placeholder to a message template and subsequent messages sent using that template will automatically include the placeholder and so have user tracking enabled.

## Examining user-tracking statistics per message

To examine the user-tracking statistics in their simplest format, click on the **messages** link in the sidebar and view the sent messages.

Here is a test message sent with user tracking enabled, but no users had opened the message yet:

If you view a message, you will be able to resend it to a different list

messages

templates

process queue

manage bounces

eventlog

sent draft queued

14 Messages

Listing message 1 to 5

<< < > >>

| Message info  | Status   | Action |      |      |     |      |   |   |   |   |   |                                   |
|---|--|--------|------|------|-----|------|---|---|---|---|---|-----------------------------------|
| From: Webmaster noreply@phplistpackt.com<br>Subject: Grey's Anatomy, Season 3 now on shelves!<br>Entered: 2010-10-22 09:59:23<br>Embargo: 2010-10-22 09:59:00 | Sent: 2010-10-22 10:00:19<br>Time to send: 1 secs<br><table border="1"><thead><tr><th>total</th><th>text</th><th>html</th><th>PDF</th><th>both</th></tr></thead><tbody><tr><td>4</td><td>0</td><td>4</td><td>0</td><td>0</td></tr></tbody></table> | total  | text | html | PDF | both | 4 | 0 | 4 | 0 | 0 | view<br>requeue<br>edit<br>delete |
| total   | text   | html   | PDF  | both |     |      |   |   |   |   |   |                                   |
| 4   | 0  | 4      | 0    | 0    |     |      |   |   |   |   |   |                                   |

Here is the same message after two users have opened the message (and at least one of them opened it more than once, given the difference between the total versus unique views!):

If you view a message, you will be able to resend it to a different list

messages

templates

process queue

manage bounces

eventlog

sent draft queued

14 Messages

Listing message 1 to 5

<< < > >>

| Message info  | Status  | Action |      |      |     |      |   |   |   |   |   |        |   |              |   |                                   |
|---|---|--------|------|------|-----|------|---|---|---|---|---|--------|---|--------------|---|-----------------------------------|
| From: Webmaster noreply@phplistpackt.com<br>Subject: Grey's Anatomy, Season 3 now on shelves!<br>Entered: 2010-10-22 09:59:23<br>Embargo: 2010-10-22 09:59:00 | Sent: 2010-10-22 10:00:19<br>Time to send: 1 secs<br><table border="1"><thead><tr><th>total</th><th>text</th><th>html</th><th>PDF</th><th>both</th></tr></thead><tbody><tr><td>4</td><td>0</td><td>4</td><td>0</td><td>0</td></tr></tbody></table><br><table border="1"><tbody><tr><td>Viewed</td><td>7</td></tr><tr><td>Unique Views</td><td>2</td></tr></tbody></table> | total  | text | html | PDF | both | 4 | 0 | 4 | 0 | 0 | Viewed | 7 | Unique Views | 2 | view<br>requeue<br>edit<br>delete |
| total   | text  | html   | PDF  | both |     |      |   |   |   |   |   |        |   |              |   |                                   |
| 4   | 0   | 4      | 0    | 0    |     |      |   |   |   |   |   |        |   |              |   |                                   |
| Viewed  | 7   |        |      |      |     |      |   |   |   |   |   |        |   |              |   |                                   |
| Unique Views  | 2   |        |      |      |     |      |   |   |   |   |   |        |   |              |   |                                   |

User Tracking Statistics

## Examining user-tracking statistics per user

In addition to examining user-tracking statistics per message, we can also view the individual actions of each user. To examine user-tracking statistics per-user, view the user's details and click on the **history** link:

**phplist - details of a user**

good morning admin

logout

English

[back to the list of users](#)

Delete will delete user and all listmemberships

delete

henrygray@phplistpackt.com [update page](#) [unsubscribe page](#) [history](#)

**User Details**

|                              |                            |
|------------------------------|----------------------------|
| ID                           | 16                         |
| Email                        | henrygray@phplistpackt.com |
| Is this user confirmed (1/0) | <input type="checkbox"/>   |

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- manage bounces
- eventlog

Now that user tracking is enabled, we are able to see the date and time at which this user opened each message and calculate a per-message, as well as an average, response time (in seconds).

**phplist - history of a user**

good morning admin

logout

English

User **henrygray@phplistpackt.com**

3 messages sent to this user

| Messages | sent                 | bounce | viewed               | responsetime |
|----------|----------------------|--------|----------------------|--------------|
| 25       | 22 Oct 2010 09:40:11 |        |                      |              |
| 26       | 22 Oct 2010 09:42:22 |        |                      |              |
| 27       | 22 Oct 2010 10:00:19 |        | 22 Oct 2010 10:03:22 | 183          |
| average  |                      |        |                      | 183          |

close

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- manage bounces
- eventlog

## Click tracking

Now that we know how to track basic message open statistics, we may want to know who opens our messages and actually clicks on the links in the message.

Unlike user tracking, click tracking is enabled / disabled globally in the `config` file and will work with text or HTML e-mails. phpList performs click tracking by rewriting every URL in your message, effectively placing itself "in between" the user and the final destination.

This means that if you add a link to `www.mywonderfulwidgets.com` to your e-mail, it would normally look like this in the e-mail source:

```
<a href="http://www.mywonderfulwidgets.com">widget</a>
```

When click tracking is enabled, the link will be rewritten like this:

```
<a href="url-to-your-phplist-install.com/lt.php?id=fk0ETwdVG1cH">widget</a>
```

phpList records a unique ID (see the rewritten link) for each user and link, meaning that it's able to redirect the rewritten link to the original link, as well as record which user clicked on the link:



Important note regarding availability of your phpList installation:  
Once click tracking is enabled, your phpList installation must be world-reachable for users to click the links and be directed to their ultimate destination.  
That is, if your phpList server is down or not publically available, your readers will be unable to click the links in your e-mail.

## Avoiding false spam / phishing detection

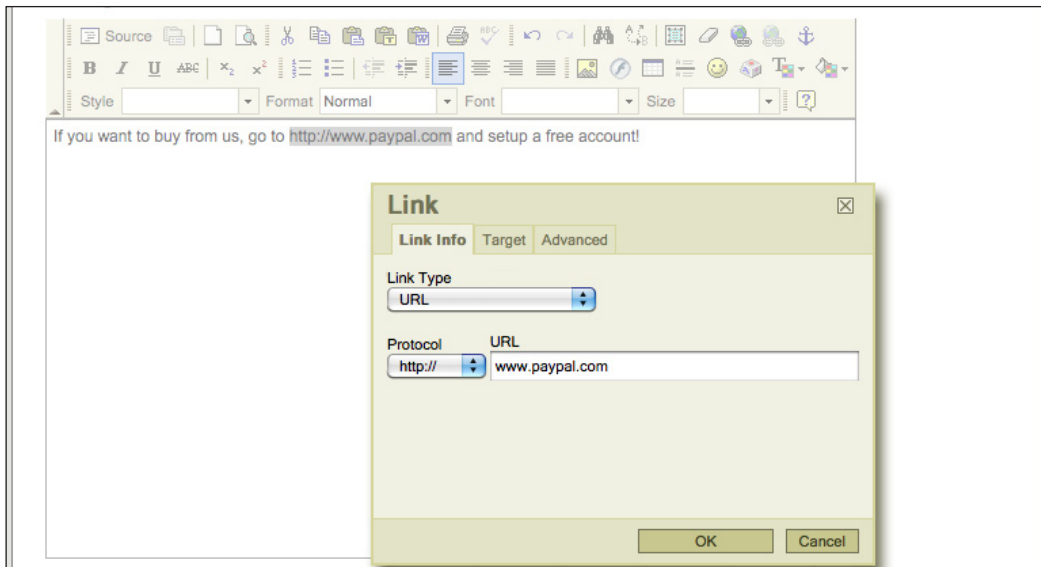
A note of caution – some e-mail systems will flag or even block e-mails which appear to contain "fraudulent" links. For example, a common technique used in "phishing" attacks is to present a link with display-text which looks legitimate to a user, such as `http://www.paypal.com`, but which, in actual fact, redirects to a malicious website (that is, `http://www.imabadguygonnastealyourmoney.com`).

In the HTML code, this would appear as follows:

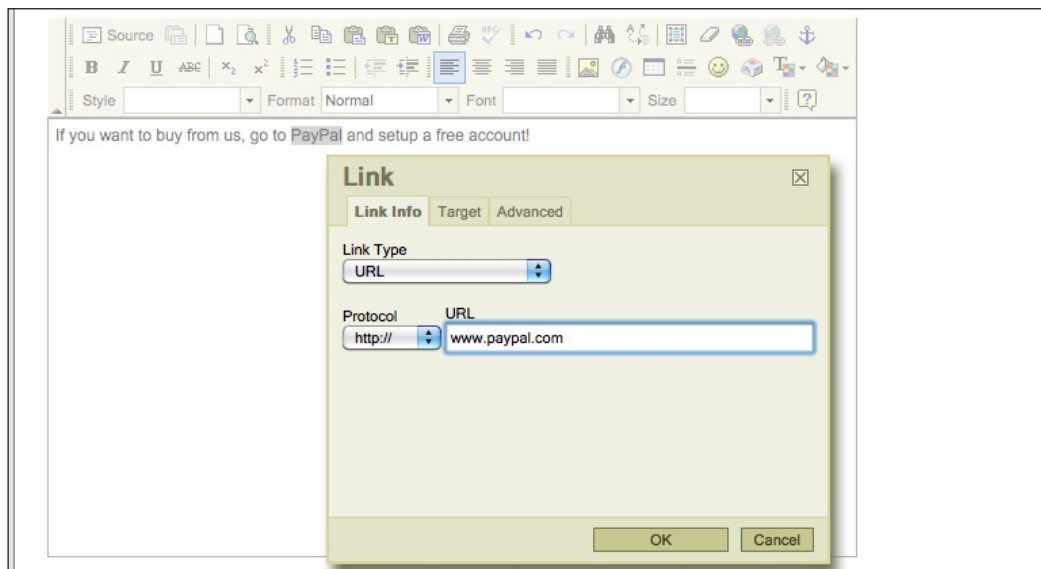
```
Go to <A HREF="http:// http://www.imabadguygonnastealyourmoney.com">www.paypal.com</A> to verify your account
```

phpList's click tracking works in a similar way. We are rewriting a standard URL to our own and then redirecting the user. To avoid being flagged as a potential "phisher", use a "friendly" display-text for your URL. That is, make sure all hyperlinks that you include in your messages have a display-text that doesn't resemble a URL.

For example, this is a bad way to include a link to `http://www.paypal.com`, because it will ultimately look like a phishing attack, once phpList has rewritten `http://www.paypal.com` to its own URL:



If we set our display-text to be simply "PayPal", then we avoid committing this (innocent) mistake:



## Enabling click tracking

To enable click tracking, open `config/config.php` and search for `CLICKTRACK`. You'll find this line:

```
define('CLICKTRACK', 0);
```

Enable the click tracking feature by changing the 0 to a 1 as follows:

```
define('CLICKTRACK', 1);
```

## Enabling extra detail in click tracking

To enable additional detail in your click tracking reports, open `config/config.php` and search for `CLICKTRACK_SHOWDETAIL` (around line #433). You'll find this line:

```
define('CLICKTRACK_SHOWDETAIL', 0);
```

Enable additional click tracking details by changing the 0 to a 1 as follows:

```
define('CLICKTRACK_SHOWDETAIL', 1);
```

Note that the commented lines in the `config` file explain the purpose of the additional detail, but we recommend against using this feature because it impacts the speed at which statistics are displayed:

**"if you enable this, you will get some extra statistics about unique users who have clicked the links in your messages and the breakdown between clicks from text or html messages. However, this will slow down the process to view the statistics, so it is recommended to leave it off, but if you're very curious, you can enable it"**

## Sending an e-mail with click tracking enabled

Because click tracking is enabled globally, there is nothing more to be done when you send a message. Simply include some links in your message to have these links rewritten for click tracking.

## Examining basic click-tracking statistics per message

As with the user tracking features, for a basic overview of the click-tracking statistics per-message, click on the **Sent** tab in the **messages** page. You'll notice an additional counter indicating how many clicks are recorded against each message.

There's also an additional action in the left-most column entitled **click stats**. This will show you detailed statistics on each URL in the message:

The screenshot shows the 'Sent' tab selected, with 18 messages listed. The message details for '3rd email about wonderful widgets' are displayed. The 'Status' section includes a table for delivery statistics and a 'Click summary' box. The 'Click summary' box shows 3 clicks. A blue box highlights the 'Click summary' box, and a blue arrow points to the 'click stats' link in the 'Action' column.

| Message info   | Status  | Action |      |      |     |      |   |   |   |   |   |  |
|--|---|--------|------|------|-----|------|---|---|---|---|---|--|
| <b>From:</b> Webmaster noreply@phplistpackt.com<br><b>Subject:</b> 3rd email about wonderful widgets<br><b>Entered:</b> 2010-10-22 11:04:56<br><b>Embargo:</b> 2010-10-22 11:04:00 | <b>Sent:</b> 2010-10-22 11:05:43<br><b>Time to send:</b> 10 secs<br><table border="1"> <thead> <tr> <th>total</th> <th>text</th> <th>html</th> <th>PDF</th> <th>both</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> </tbody> </table><br><b>Viewed</b> 2<br><b>Unique Views</b> 2<br><b>Clicks</b> 3 | total  | text | html | PDF | both | 4 | 0 | 4 | 0 | 0 | <a href="#">view</a><br><a href="#">requeue</a><br><a href="#">edit</a><br><a href="#">click stats</a><br><a href="#">delete</a> |
| total  | text  | html   | PDF  | both |     |      |   |   |   |   |   |  |
| 4  | 0   | 4      | 0    | 0    |     |      |   |   |   |   |   |  |

Clicking on the **view users** link will show you which users clicked on this URL in this particular message:

The screenshot shows the 'Click Details for a Message' page. The message details are shown at the top. Below is a table of 'Message Click Statistics' for the URL 'http://www.mywonderfulwidgets.com'. The table shows 3 clicks, a click rate of 75.00%, and the first click occurred at 11:05:57 and the latest at 11:06. A blue arrow points to the 'view users' link in the 'who' column.

| Message Click Statistics          | firstclick           | latestclick       | clicks | Clickrate | who                        |
|-----------------------------------|----------------------|-------------------|--------|-----------|----------------------------|
| http://www.mywonderfulwidgets.com | 22 Oct 2010 11:05:57 | 22 Oct 2010 11:06 | 3      | 75.00%    | <a href="#">view users</a> |
| total                             |                      |                   | 3      |           |                            |



A list of all the users who clicked on the URL is displayed. Note that it's possible for a user to click a link several times; that is why a **firstclick** and a **latestclick** entry are recorded for each:

| User Click Details for a URL in a message <a href="http://www.mywonderfulwidgets.com">http://www.mywonderfulwidgets.com</a> |                                   |                   |        |
|---|-----------------------------------|-------------------|--------|
| Subject   | 3rd email about wonderful widgets |                   |        |
| Entered   | 2010-10-22 11:04:56               |                   |        |
| Sent  | 2010-10-22 11:05:43               |                   |        |
| User Click Statistics   | firstclick                        | latestclick       | clicks |
| <a href="mailto:theodorbillroth@phplistpackt.com">theodorbillroth@phplistpackt.com</a>                                      | 22 Oct 2010 11:06:01              | 22 Oct 2010 11:06 | 2      |
| <a href="mailto:josephlister@phplistpackt.com">josephlister@phplistpackt.com</a>  | 22 Oct 2010 11:05:57              | 22 Oct 2010 11:05 | 1      |
| total   |                                   |                   | 3      |
| <a href="#">close</a> ↗   |                                   |                   |        |

[send a message](#)  
[users](#)  
[manage users](#)  
[subscribe pages](#)  
[messages](#)  
[statistics](#)  

---

[templates](#)  
[process queue](#)  
[manage bounces](#)  
[eventlog](#)

## Examining advanced user/click-tracking statistics

While we can examine both basic-user and click-tracking statistics from the **Sent** tab on the **messages** page, we can also use the new **statistics** page to give us more details.

### Extra statistics sidebar link

Once click tracking is enabled, an additional link will appear on your sidebar labeled **statistics**:

|                                |  |
|--------------------------------|--|
| <b>System Functions</b>        |  |
| <a href="#">setup</a>          | <a href="#">Setup phplist</a>            |
| <a href="#">dbcheck</a>        | <a href="#">Check Database structure</a> |
| <a href="#">eventlog</a>       | <a href="#">View the eventlog</a>        |
| <a href="#">close</a> ↗        |  |
| <b>Configuration functions</b> |  |

[send a message](#)  
[users](#)  
[manage users](#)  
[subscribe pages](#)  
[messages](#)  
[statistics](#)  

---

[templates](#)  
[process queue](#)

This link will take you to a page (under development) enabling more detailed statistics about user and link tracking:

|  |   |
|--|---|
| <p>This page shows the available statistics for retrieval. It is currently still a bit rough and will need updating in future versions.</p> <p>Post your comments and suggestions to <a href="#">Mantis</a></p> <ul style="list-style-type: none"> <li>overview</li> <li>view clicks by url</li> <li>view clicks by message</li> <li>view opens by message</li> <li>domain statistics</li> </ul> | <ul style="list-style-type: none"> <li>send a message</li> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> <li>statistics</li> </ul> <hr/> <ul style="list-style-type: none"> <li>templates</li> <li>process queue</li> <li>manage bounces</li> </ul> |
|--|---|

## Overview

The **overview** page is a useful place to start. It will show you your most recent messages and indicate how many were sent, how many bounced, and how many were viewed:

| Select Message to view                                      |             |         |       |      |         |
|---|-------------|---------|-------|------|---------|
| Last few Messages   | date sent   | bounced | views | rate |         |
| <a href="#">30 Email with click-tracking enabled</a>        | 22 Oct 2010 | 4       | 0     | 1    | 25.00 % |
| <a href="#">28 Non-click-tracked email</a>                  | 22 Oct 2010 | 4       | 0     | 0    | 0.00 %  |
| <a href="#">27 Grey's Anatomy, Season 3 now on shelves!</a> | 22 Oct 2010 | 4       | 0     | 2    | 50.00 % |
| <a href="#">26 Grey's Anatomy, Season II</a>                | 22 Oct 2010 | 4       | 0     | 3    | 75.00 % |
| <a href="#">25 Buy your medical supplies</a>                | 22 Oct 2010 | 4       | 0     | 0    | 0.00 %  |

The **viewed** counter is a link to the message detail page, which will show you which users viewed the message:

| View Details for a Message  |                             |                         |              |
|---|-----------------------------|-------------------------|--------------|
| Subject   | Grey's Anatomy, Season II   |                         |              |
| Entered   | 2010-10-22 09:41:27         |                         |              |
| sent  | 2010-10-22 09:42:23         |                         |              |
| 3 Entries   |                             |                         |              |
| Listing user 1 to 50  |                             |                         |              |
| <input type="button" value="&lt;&lt;"/> <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&gt;&gt;"/> |                             |                         |              |
| Message Open Statistics   | sent                        | firstview               | responsetime |
| <a href="#">frederickbanting@phplistpackt.com</a>   | 22 October 2010<br>09:42:22 | 22 Oct 2010<br>09:55:26 | 784 sec      |
| <a href="#">theodorbillroth@phplistpackt.com</a>  | 22 October 2010<br>09:42:22 | 22 Oct 2010<br>10:01:22 | 1140 sec     |
| <a href="#">josephlister@phplistpackt.com</a>   | 22 October 2010<br>09:42:23 | 22 Oct 2010<br>09:55:32 | 789 sec      |
| <a href="#">close</a> ↗   |                             |                         |              |

## Viewing clicks by URL

From the **statistics** page, the **view clicks by url** link will display all the URLs you've used in all your messages. On this page, the total clicks per URL will be indicated:

| Select URL to view  |        |
|---|--------|
| Available URLs  | clicks |
| <a href="http://www.mywonderfulwidgets.com">http://www.mywonderfulwidgets.com</a> | 4      |
| <a href="http://mywonderfulwidgets.com">http://mywonderfulwidgets.com</a>         | 1      |
| <a href="#">close</a> ⬆   |        |

[send a message](#)  
[users](#)  
[manage users](#)  
[subscribe pages](#)  
[messages](#)  
[statistics](#)

Clicking on the URL will take you to the URL detail page, which will show more detailed statistics for each message:

| Click Details for a URL <a href="http://www.mywonderfulwidgets.com">http://www.mywonderfulwidgets.com</a> |                      |                   |        |           |                            |
|---|----------------------|-------------------|--------|-----------|----------------------------|
| URL Click Statistics  | firstclick           | latestclick       | clicks | clickrate | who                        |
| msg 30: Email with click-tracking   | 22 Oct 2010 10:34:03 | 22 Oct 2010 10:34 | 1      | 12.50%    | <a href="#">view users</a> |
| msg 32: 3rd email about wonderful   | 22 Oct 2010 11:05:57 | 22 Oct 2010 11:06 | 3      | 37.50%    | <a href="#">view users</a> |
| total   |                      |                   | 4      |           |                            |
| <a href="#">close</a> ⬆   |                      |                   |        |           |                            |

[send a message](#)  
[users](#)  
[manage users](#)  
[subscribe pages](#)  
[messages](#)  
[statistics](#)  

---

[templates](#)  
[process queue](#)  
[manage bounces](#)

Clicking on the **message** link will display the click statistics for this message and clicking on the **view users** link will show which users clicked this link in this particular message:

| User Click Details for a URL in a message <a href="http://www.mywonderfulwidgets.com">http://www.mywonderfulwidgets.com</a> |                      |                   |        |  |
|---|----------------------|-------------------|--------|--|
| Subject <a href="#">3rd email about wonderful widgets</a>   |                      |                   |        |  |
| Entered 2010-10-22 11:04:56   |                      |                   |        |  |
| Sent 2010-10-22 11:05:43  |                      |                   |        |  |
| User Click Statistics   | firstclick           | latestclick       | clicks |  |
| <a href="mailto:theodorbillroth@phplistpackt.com">theodorbillroth@phplistpackt.com</a>                                      | 22 Oct 2010 11:06:01 | 22 Oct 2010 11:06 | 2      |  |
| <a href="mailto:josephlister@phplistpackt.com">josephlister@phplistpackt.com</a>  | 22 Oct 2010 11:05:57 | 22 Oct 2010 11:05 | 1      |  |
| total   |                      |                   | 3      |  |
| <a href="#">close</a> ⬆   |                      |                   |        |  |

[send a message](#)  
[users](#)  
[manage users](#)  
[subscribe pages](#)  
[messages](#)  
[statistics](#)  

---

[templates](#)  
[process queue](#)  
[manage bounces](#)  
[eventlog](#)

## Viewing clicks by message

The **view clicks by message** link on the **statistics** page will display click statistics on a per-message basis (instead of the per-URL basis, as before):

| Select Message to view                      |        |       |       |        | <a href="#">send a message</a><br><a href="#">users</a><br><a href="#">manage users</a><br><a href="#">subscribe pages</a><br><a href="#">messages</a><br><a href="#">statistics</a><br><hr/> <a href="#">templates</a> |
|---|--------|-------|-------|--------|---|
| Available Messages                          | clicks | users | links | rate   |   |
| <b>32</b> 3rd email about wonderful widgets | 3      | 2     | 2     | 0.00 % |   |
| <b>31</b> Another email about widgets       | 1      | 1     | 1     | 0.00 % |   |
| <b>30</b> Email with click-tracking enabled | 1      | 1     | 1     | 0.00 % |   |
| <a href="#">close</a> ↗                     |        |       |       |        |   |

Clicking on a message will then show statistics about all the URLs included in that message:

| Click Details for a Message   |                         |                      |        |           |   | <a href="#">send a message</a><br><a href="#">users</a><br><a href="#">manage users</a><br><a href="#">subscribe pages</a><br><a href="#">messages</a><br><a href="#">statistics</a><br><hr/> <a href="#">templates</a><br><a href="#">process queue</a><br><a href="#">manage bounces</a><br><a href="#">eventlog</a> |
|---|-------------------------|----------------------|--------|-----------|---|--|
| Subject 3rd email about wonderful widgets   |                         |                      |        |           |   |  |
| Entered 2010-10-22 11:04:56   |                         |                      |        |           |   |  |
| Sent 2010-10-22 11:05:43  |                         |                      |        |           |   |  |
| Message Click Statistics  | firstclick              | latestclick          | clicks | Clickrate | who   |  |
| <a href="http://www.mywonderfulwidgets.com">http://www.mywonderfulwidgets.com</a> | 22 Oct 2010<br>11:05:57 | 22 Oct 2010<br>11:06 | 3      | 75.00%    | <a href="#">view</a><br><a href="#">users</a> |  |
| total   |                         |                      | 3      |           |   |  |
| <a href="#">close</a> ↗   |                         |                      |        |           |   |  |

## Viewing opens by message

The link to **view opens by message** displays the same data as the **overview** link, except the messages' subjects are not clickable, so this page essentially provides **less** data than the overview page. Let's ignore it.

## Viewing domain statistics

This seems to be part of the ongoing development of phpList. At the time of writing, this link displays nothing. Let's ignore it too.

## Summary

In this chapter, you have learned about user and click tracking, specifically how they operate, and the limitations of each. You've examined user-and click-tracking statistics to determine which users are opening which messages and which links are being clicked. Specifically, you've covered:

- Enabling user tracking and viewing statistics about who's reading our messages
- Enabling click tracking and viewing statistics about who's clicking on which links in our messages
- Examining user/click statistics in greater detail using the "statistics" menu

In the following chapter, we'll be looking at securing your phpList installation, both against common attacks (that is, brute force) and against potential (current and future) vulnerabilities.

# 8

## Securing phpList

Having a fully functioning e-mail delivery system, we now want to ensure that we secure it as best we can, namely, by preventing both unauthorized access and exposure to potential vulnerabilities.

In this chapter, you will learn some techniques to further restrict and secure your phpList installation. These include:

- Changing the administrator password
- Confirming `.htaccess mod_access` restrictions
- Securing the admin pages with an additional (`.htaccess mod_auth`) password
- Confirming appropriate filesystem permissions

### Changing the admin password

Your password is the "key" to your phpList installation. Just as you wouldn't leave your house keys outside the front door, or your PIN number written onto your bank cards, security begins with effective password management.

### Choosing strong passwords

A strong password is one that is difficult to detect by humans or computers. It is generally accepted that a strong password:

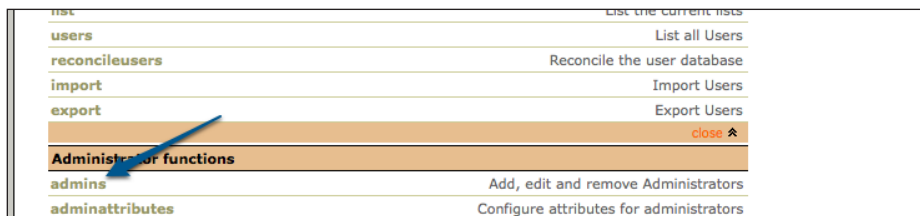
- Is unique to this application (that is, don't use the same password that you use elsewhere).
- Is at least eight characters long.
- Contains numbers and letters in both upper and lower case. Consider using symbols too.

There are several websites that will help generate / validate strong passwords. (Just search the web for "how strong is my password?").

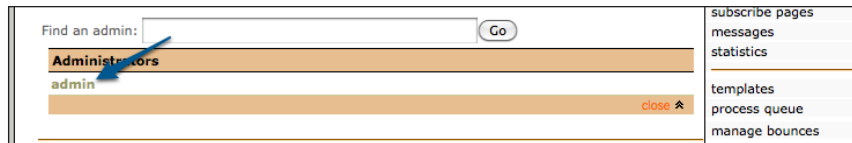
## Changing phpList admin password

Hopefully, you'll already have changed the "admin" password from the default "phplist", but make sure it's set to a strong password that won't be guessable (that is, if your domain is "fuzzyslippers.com", don't make the password "fuzzyslippers!").

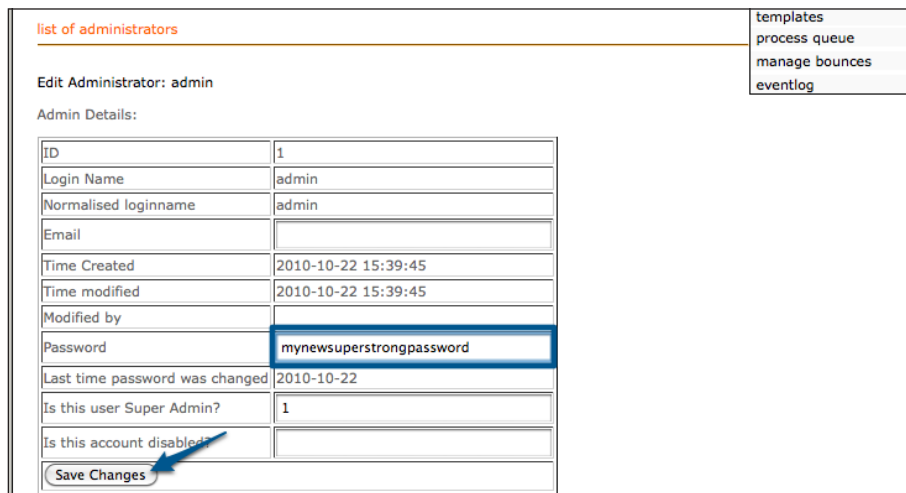
To change the admin password, click on the **admins** link on the main page:



Click on your **admin** account (you can have multiple administrators):



Change your password (displayed in clear text) and click on **Save Changes**:



## Confirming correct filesystem permissions

Every file on a Unix-based filesystem has three levels of access permission – read, write, and execute. These permissions are assigned to three separate entities – the owner, a user group, and the world. Best practice dictates that these permissions be set as low (restrictive) as possible (that is, unless you really NEED world-writeable files, don't set them that way!).

It's important to ensure that the filesystem permissions for all of the phpList's files are correct. If the permissions are too restrictive, the web server won't be able to display the phpList interface at all. If they are too loose, you risk unauthorized users editing or deleting your files.

If you unpacked phpList onto your web host directly from the source `tar.gz` file, then the permissions will be correctly set. However, if you transferred the files from another host using FTP, it's possible that the permissions will have been reset. Regardless, it's advisable to inspect these.

## Confirming permissions in a GUI SFTP/FTP client

Most users will interact with their web host via a GUI SFTP/FTP client. Your client will generally show you the permissions on each file using the `drwxrwxrwx` syntax or in numeric shorthand called "octal notation":

| Filename   | ▲ Size  | Modified          | Permissions                  |
|------------|---------|-------------------|------------------------------|
| admin      | 4.0 KB  | 22/10/10 2:57 PM  | <code>rwxr-xr-x (755)</code> |
| config     | 4.0 KB  | 22/10/10 12:36 PM | <code>rwxr-xr-x (755)</code> |
| dl.php     | 2.2 KB  | 6/05/09 1:13 AM   | <code>rw-r--r-- (644)</code> |
| images     | 4.0 KB  | 24/04/10 3:05 AM  | <code>rwxr-xr-x (755)</code> |
| index.html | 698 B   | 15/04/04 8:40 PM  | <code>rw-r--r-- (644)</code> |
| index.php  | 35.0 KB | 25/09/10 3:18 PM  | <code>rw-r--r-- (644)</code> |
| js         | 4.0 KB  | 24/04/10 3:05 AM  | <code>rwxr-xr-x (755)</code> |
| lists      | 4.0 KB  | 25/09/10 10:35 AM | <code>rwxr-xr-x (755)</code> |
| lt.php     | 3.7 KB  | 6/05/09 1:13 AM   | <code>rw-r--r-- (644)</code> |
| styles     | 4.0 KB  | 24/04/10 3:05 AM  | <code>rwxr-xr-x (755)</code> |
| test.html  | 1.6 KB  | 25/09/10 2:33 PM  | <code>rw-r--r-- (644)</code> |
| test.php   | 318 B   | 25/09/10 1:25 PM  | <code>rw-r--r-- (644)</code> |
| texts      | 4.0 KB  | 24/04/10 3:05 AM  | <code>rwxr-xr-x (755)</code> |
| ut.php     | 1.8 KB  | 6/05/09 1:13 AM   | <code>rw-r--r-- (644)</code> |



## Confirming permissions using a shell session

The same information can be displayed if you have a shell on the web server host using the `ls -l` command:

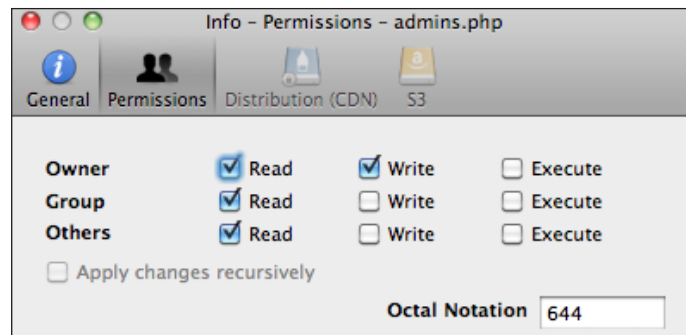
```
[root@myhost public_html]# ls -l
total 88
drwxr-xr-x 15 root root 4096 Oct 22 14:57 admin
drwxr-xr-x  2 root root 4096 Oct 22 12:36 config
-rw-r--r--  1 root root 2283 May  6 2009 dl.php
drwxr-xr-x  3 root root 4096 Apr 24 03:05 images
-rw-r--r--  1 root root  698 Apr 15 2004 index.html
-rw-r--r--  1 root root 35860 Sep 25 15:18 index.php
drwxr-xr-x  2 root root 4096 Apr 24 03:05 js
drwxr-xr-x  2 root root 4096 Sep 25 10:35 lists
-rw-r--r--  1 root root 3873 May  6 2009 lt.php
drwxr-xr-x  2 root root 4096 Apr 24 03:05 styles
-rw-r--r--  1 root root 1683 Sep 25 14:33 test.html
-rw-r--r--  1 root root  318 Sep 25 13:25 test.php
drwxr-xr-x  2 root root 4096 Apr 24 03:05 texts
-rw-r--r--  1 root root 1847 May  6 2009 ut.php
[root@myhost public_html]#
```

## Setting appropriate permissions

Ensure that all directories are set to `rwxr-xr-x` or `755` (read, write, and execute access for the owner, read and execute for the group and world) and that all files are set to `rw-r--r--`, or `644` (read and write access for the owner, read access for the group and world).

On a Unix shell, you do this using the `chmod` command. That is, to change the permissions on a file to `644`, you'll run `chmod 644 filename`.

In your GUI client, there's likely a context menu item to set permissions on a selection of files:



## Confirming htaccess restrictions are in place

phpList ships with a preconfigured `.htaccess` file (distributed Apache configuration directives) which protects certain files in the `admin/` subfolder from being processed individually (most of these files are intended to be referenced within `index.php`, not directly).

Certain web hosting providers may not allow `.htaccess` files to alter the web server behavior. To test whether your `admin/.htaccess` file is effective, try to open the URL `http://url-to-your-phpList-install.com/admin/subscribeLib2.php` in your browser.

If your `.htaccess` file is working as expected, you will receive a **403 Forbidden** message, as this file is not intended to be accessed directly.

However, if your `.htaccess` file is ineffective, you will see a single line, reading **Invalid Request**. If this is the case, then your `.htaccess` file is being ignored by your web server and you should contact your hosting provider to address this. Your phpList installation is vulnerable.

## Securing admin pages with additional (htauth) password

By default, to view the admin pages, you need to enter an administrator username and password on the phpList admin login page. However, this is only a single layer of security. If your phpList database was to be compromised, or the routines used to validate passwords were discovered to be exploitable, an attacker could gain full control over your phpList installation.

To add an additional layer of security, we can configure an additional Apache `mod_auth`-based password to protect the admin page. This means that your web server (prior to phpList) will prompt the viewer for a user and password combination, before showing them the phpList login page.



This also means that you need two passwords to administer your list!

## Creating an htpasswd file

The `mod_auth` authentication scheme requires a file commonly called an *htpasswd* file, which contains pairs of usernames and encrypted passwords, one per line, to exist on the web server. The web server will use this file to validate your username and password when you browse to any URL underneath the `/admin/` directory.

This file should be accessible to the web server (that is, on the web server's filesystem), but not accessible to the world (that is, not in a directory which is accessible via a web browser).

In many cases, web-facing files (such as phpList's `.php` files) are stored in a `public_html` folder on your web host. You may have other folders including `log` files, `backups`, and so on. Choose a location for the password file and make sure you know the filesystem path to this file. (For example, `/home/webusers/myusername/htpasswd` would be secure, if your web files are in `/home/webusers/myusername/public_html/`).

## Creating an htpassword file online

The simplest way to generate an `htpassword` file is to generate the user-encrypted password pairs online. There are many online tools that use simple code to do this for you. You could search the web for "generate htpasswd file" or use a website like <http://www.htaccesstools.com/htpasswd-generator/>.

Once you have the user-encrypted password line, save it to a file named `htpasswd`, and upload it to the chosen directory on your web host. Remember, you need to know the filesystem path to this file.

## Creating an htpasswd file in a Unix shell

Assuming you have access to your web host using a Unix shell (or to another Unix-like host), you can also create the password file directly on the web host, using the following command:

```
htpasswd -cb <filesystem path to password file> <username> <password>
```

For example, you may use:

```
[root@webhost myuser]# htpasswd -cb /home/webusers/myusername/passwd
myuser mypassword
```

Adding password for user myuser

```
[root@webhost myuser]#
```



The `-c` in the `htpasswd` command stands for "create". It assumes that the password file doesn't already exist. If you run this against an existing file, it will overwrite this file, so be careful! If you want to add additional usernames to this file, use the `htpasswd` command without the `-c` argument.

The contents of your password file should look something like this: `myuser:CEZqTfQTUZhzw`.

## Amending admin/.htaccess

Having a file containing usernames and passwords is not enough to secure access—we need to tell the web server not to allow any visits to any URLs underneath the `admin/` directory, unless the user has authenticated against our password file.

We put these instructions in a special file in the `admin/` directory called `.htaccess`.


By default, this file contains the following lines, which restrict any access to important `.php` files under the `admin/` directory:

```
<FilesMatch "\.(php|inc)$">
Order allow,deny
deny from all
</FilesMatch>
<FilesMatch "(index.php|connector.php|upload.php)$">
Order allow,deny
allow from all
</FilesMatch>
```

We will add the following directives to this file either above or below the existing text:

```
# Restrict access to admin/ URLs
AuthUserFile <filesystem path to your htpasswd file>
AuthName "PHPList Admin Access"
AuthType Basic
require valid-user
# End access restriction directives
```

These lines tell the web server which password file to refer to, what the name of the authentication "realm" is, what type of authentication to use, and instruct it that a valid user match is required.

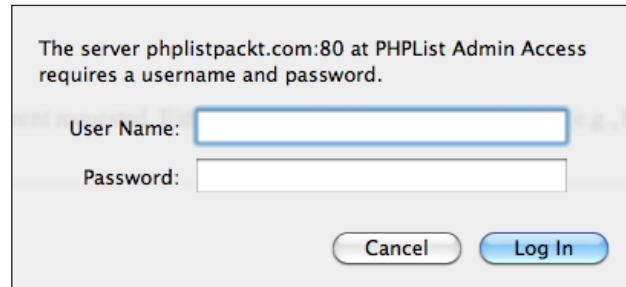
[  **Comments** ]

The lines starting with hash characters are comments and are included simply to improve readability. They can be excluded or customized and are ignored by the web server.

Having added the aforementioned lines, the entire admin/ .htaccess file will look something like this:

```
<FilesMatch "\.(php|inc)$">
Order allow,deny
deny from all
</FilesMatch>
<FilesMatch "(index.php|connector.php|upload.php)$">
Order allow,deny
allow from all
</FilesMatch>
# Restrict access to admin/ URLs
AuthUserFile <filesystem path to your htpasswd file>
AuthName "PHPList Admin Access"
AuthType Basic
require valid-user
# End access restriction directives
```

Having saved the changes to `admin/.htaccess`, refresh the admin page and your browser will pop up an authentication request as shown in the following screenshot:



The screenshot shows a dialog box with a light gray background. At the top, it says "The server phplistpackt.com:80 at PHPList Admin Access requires a username and password." Below this text are two input fields: "User Name:" followed by a white text box with a blue border, and "Password:" followed by a white password box with a blue border. At the bottom of the dialog, there are two buttons: a "Cancel" button on the left and a "Log In" button on the right.



### Remembering passwords

Some web browsers will offer you the option of remembering this password for the future, which is suitable if you are using a browser on a trusted, private computer (that is, don't do this at an Internet cafe!).



Note that using this method of restricting access doesn't allow you any means to log out again. Once you enter your username and password, your browser will keep you logged in (at the `mod_auth` level) until you restart your browser.

## Summary

In this chapter, you've learned some simple, and some advanced, techniques to secure your phpList installation, including:

- Changing the administrator password
- Confirming `.htaccess` `mod_access` restrictions
- Securing the admin pages with an additional (`.htaccess` `mod_auth`) password
- Confirming the appropriate filesystem permissions

In the next chapter, we'll look at using some of phpList's more advanced features and settings.



# 9

## Advanced Features

In *Chapter 7, Measuring Effectiveness of Newsletters*, we talked about measuring the effectiveness of your newsletters and we covered manually enabling more advanced click-tracking features by editing the `config/config.php` file. In this chapter, we'll cover some of the more advanced features of phpList including:

- Advanced bounce handling
- Domain-based e-mail throttling
- Adding attachments to messages
- Automatically repeating messages
- Including RSS feeds in messages
- Setting up processing automation

### Advanced bounce handling

If we have a high volume of users, lists, and bounces, we might want to configure advanced bounce handling to better automate the processing of bounces.



Note that this is classified as an experimental advanced phpList feature.

To enable advanced bounce handling, open `config/config.php` and look for this line:

```
define('USE_ADVANCED_BOUNCEHANDLING', 0);
```




Change `USE_ADVANCED_BOUNCEHANDLING` from 0 to 1 and then reload the admin page. You'll note that the **view bounces** link has been replaced with **manage bounces**:



## Bounce rules and regular expressions

Advance bounce handling lets us configure different actions for incoming bounce messages based on regular expressions.

[  **Regular expressions** Regular expressions are extremely powerful, but in our examples, we'll just use simple string matching. You can find a detailed description of what they are and how to use them at [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression). ]

## Creating a new bounce rule

To create a new bounce rule (from scratch), click on **list bounce rules** to display all rules (by default, there are none) and fill out the following under the **add a new rule** section:

- **Regular expression:** The text match to perform against the bounce message
- **Action:** A specific action to perform based on this bounce
- **Memo for this rule:** A note to yourself to explain what this rule does and why. (It's fairly obvious in this example, but more complex regular expressions may require explanation.)

powered by:  
[ PHP ] + [ MySQL ]

phplist

logout

main page

configure

help

about

logout

English

active candidate

No Rules found

add a new rule

Regular Expression: user account suspended for abuse

action: blacklist user

Memo for this rule

If email bounces because user has been suspended for abuse, then blacklist them from our system.

Add new Rule

lists

send a message

users

manage users

subscribe pages

messages

templates

process queue

manage bounces

eventlog

## Creating a new rule based on an existing bounce

Instead of creating new rules from scratch, it's more likely that we'll need to create new rules from existing bounces as the need arises.

Now that we've enabled advanced bounce processing, we have an additional option when viewing a bounce (using **manage bounces** and then **view bounces**) to create a new rule based on the bounce:

powered by:  
[ PHP ] + [ MySQL ]

phplist  
logout  
main page  
configure  
help  
about  
logout  
English  
lists  
send a message  
users  
manage users  
subscribe pages  
messages  
templates  
process queue  
manage bounces  
eventlog

### phplist - view a bounce

good morning admin

Possible Actions:

For user with email

Increase bouncecount with  (use negative numbers to decrease)

Mark user as unconfirmed  (so you can resend the request for confirmation)

Set user to receive text instead of HTML

Delete user

Delete this bounce and go to the next

[Create New Rule based on this bounce](#)

Bounce Details:

|         |                          |
|---------|--------------------------|
| ID      | 15                       |
| Date    | 2010-10-09 11:07:00      |
| Status  | bounced list message 24  |
| Comment | 13 bouncecount increased |

This simply gives us the convenient view of the actual bounce body above the new rule creation dialog so that we can easily create (or copy and paste) our text match.

## Auto-generating new bounce rules

phpList can auto-generate bounce rules, based on existing bounces, if you click on the **generate bounce rules** link:

|  |   |
|--|---|
| check current bounce rules<br>process bounces<br>You have already defined bounce rules in your system. Be careful with generating new ones, because these may interfere with the ones that exist.<br><a href="#">generate bounce rules</a> | <a href="#">subscribe pages</a><br><a href="#">messages</a><br><hr/> <a href="#">templates</a><br><a href="#">process queue</a><br><a href="#">manage bounces</a> |
|--|---|

Without further prompting, phpList will review the current bounces and generate new rules based on these bounces:

|   |   |
|---|---|
| 1 new rules found<br>0 bounces not matched<br>0 bounces matched to existing rules | <a href="#">manage users</a><br><a href="#">subscribe pages</a><br><a href="#">messages</a> |
|---|---|

You can view these new "suggested" bounce rules by clicking on **list bounce rules**, then by changing the tab from **active** to **candidate**, and then clicking on the "rule number" link to edit the rule:

|   |  |
|---|--|
| Number of active rules: 1<br>Number of candidate rules: 1<br><a href="#">active</a> <a href="#">candidate</a><br><b>Bounce</b><br><b>Regular</b><br><b>Expressions</b><br>rule 9<br>http:...mail.google.com.support.bin.answer.py.answer=6596<br>o1si9286646wfl.93<br>5.1.1 blacklist<br>user<br>and<br>delete<br>bounce<br>close | <a href="#">users</a><br><a href="#">manage users</a><br><a href="#">subscribe pages</a><br><a href="#">messages</a><br><hr/> <a href="#">templates</a><br><a href="#">process queue</a><br><a href="#">manage bounces</a><br><a href="#">eventlog</a> |
| with tagged rules: <b>delete</b> <input type="checkbox"/><br><b>make active</b> <input type="checkbox"/><br><input type="button" value="Save Changes"/>   |  |

phpList would have tried to select an appropriate regular expression. Below the memo is listed the bounces which were used to derive this rule. Make any necessary changes (that is, make the regular expression as generic as possible) and change the status from **candidate** to **active** to enable the rule:

back to list of bounce rules

Regular Expression: 5.1.1 http...mail.google.com.support.bir

Created By: admin

Action: blacklist user and delete bounce

Status: candidate

Memo for this rule: Auto Created from bounce  
line: <<< 550 5.1.1 http://mail.google.com/support/bin/answer.py?answer=6596 o1si9286646w1d.93

Save Changes

related bounces: 14

Which bounces was this rule derived from?

send a message  
users  
manage users  
subscribe pages  
messages

templates  
process queue  
manage bounces  
eventlog

## Checking current rules against bounces

Having created your bounce rules, you can test them against the current bounces, by clicking on **check current bounce rules** in the **manage bounces** page:

list bounce rules

view bounces

view bounces per list

check current bounce rules

process bounces

send a message  
users  
manage users  
subscribe pages  
messages

Note that if a bounce is not matched by any bounce rules, the "default" bounce processing system will apply (un-confirming the user after consecutive bounces exceeds threshold).

## Domain-based e-mail throttling

You may want to take advantage of phpList's "domain throttling" features if you have lots of subscribers on the same domain and you want to avoid having yourself classed as a "spammer" by their e-mail provider. Throttling slows down e-mail delivery to a maximum messages-per-period rate, ensuring that you don't flood a particular e-mail host with too many concurrent messages.

To use domain throttling, edit the following lines in `config/config.php`:

```
define('USE_DOMAIN_THROTTLE', 0);
define('DOMAIN_BATCH_SIZE', 1);
define('DOMAIN_BATCH_PERIOD', 120);
```

Change `USE_DOMAIN_THROTTLE` to 1 to turn on the domain throttling feature, and then customize `DOMAIN_BATCH_SIZE` and `DOMAIN_BATCH_PERIOD` (in seconds) for the amount of e-mails (that is, `BATCH_SIZE`) to be sent to a given domain within a particular period.

For example, you may want to deliver a maximum of one message per minute to each destination domain. In this case, you'd use the following configuration:

```
define('USE_DOMAIN_THROTTLE', 1);
define('DOMAIN_BATCH_SIZE', 1);
define('DOMAIN_BATCH_PERIOD', 60);
```

It's recommended to implement this after setting up regular, automated queue process actions (we'll look at this later in this chapter), because depending on the size and diversity of your subscriber base, domain throttling may vastly increase the time it takes to send your messages to all of your subscribers.

## Adding attachments to messages

phpList supports the adding of attachments to your messages. The `config/config.php` file warns that this feature is experimental and may result in large e-mails. It also cautions that it's generally more acceptable to send a link to a file download, instead of including that file as an attachment in the message sent to the subscriber.

However, there are many situations where you might want to include attachments and doing so is quite simple.

To enable attachments, look for this line in `config/config.php`:

```
define("ALLOW_ATTACHMENTS", 0);
```

Change ALLOW\_ATTACHMENTS from 0 to 1 to turn on attachment support.

You'll note on the **send a message** page that there's an additional tab labeled **Attach**. Click on this tab while preparing your message to add attachments. The attachment tab will display the current (server-set) size limits and allow you to browse for new files to attach:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

users  
manage users  
subscribe pages  
messages  
statistics

templates  
process queue  
manage bounces  
eventlog

**Content** **Format** **Attach** **Scheduling** **Criteria** **Lists** **Misc**

? Add attachments to your message

The upload has the following limits set by the server:  
Maximum size of total data being sent to the server: 8M  
Maximum size of each individual file: 10M

New Attachment  **Browse...** **Add (and save)**

Description of attachment:

**Send Test Message** to email address(es):   
(comma separated addresses - all must be users)

**Save Changes**

It's recommended that you add a description of each attachment, although this is only for your own clarity and is not visible to your readers. Click on **Add (and save)** when you've selected the file (on your local system) to attach and entered an optional description:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

users  
manage users  
subscribe pages  
messages  
statistics

---

templates  
process queue  
manage bounces  
eventlog

---

**Content** **Format** **Attach** **Scheduling** **Criteria** **Lists** **Misc**

? Add attachments to your message

The upload has the following limits set by the server:  
Maximum size of total data being sent to the server: 8M  
Maximum size of each individual file: 10M

New Attachment:

Description of attachment:

---

to email address(es):   
(comma separated addresses - all must be users)

---

phpList will confirm the successful attachment of the file and display a summary of all files attached, including a sequence number (this will increment over time with every new attachment), a filename, description, and file size.



It's a good idea to do a "sanity-check" at this point and make sure that the file attachment won't make the message so large that it will cause inconvenience to your readers.

To add another file, repeat this process, one file at a time.

## Adding multiple files simultaneously

While phpList doesn't impose a limit on how many attachments can be added to a message, it does (by default) expect you to attach them one at a time.

You may have a requirement to attach several files to every message and find the one-at-a-time approach laborious. phpList can be configured to attach many files simultaneously.



Look for this line in `config/config.php`:

```
define ("NUMATTACHMENTS" , 1 );
```

And change `NUMATTACHMENTS` from 1 to the number you require.

For example, after changing this value to 2, you're presented with two file-attachment forms on the same page:

The screenshot shows a web interface for sending a message. At the top, there is a navigation menu with tabs: Content, Format, Attach, Scheduling, Criteria, Lists, and Misc. The 'Attach' tab is selected. Below the tabs, there is a section titled 'Add attachments to your message'. This section contains two identical forms for adding attachments. Each form has a 'New Attachment' input field, a 'Browse...' button, and an 'Add (and save)' button. Below the input field is a 'Description of attachment:' label and a text area. At the bottom of the form, there is a 'Send Test Message' button followed by an input field for email addresses, with a note '(comma separated addresses - all must be users)'. A 'Save Changes' button is located at the bottom of the page. On the right side of the interface, there is a sidebar menu with the following items: users, manage users, subscribe pages, messages, statistics, templates, process queue, manage bounces, and eventlog.



Note that clicking any of the **Add (and save)** buttons will add all attachments you've selected with the **Browse** button. The reason this button is repeated for every upload is that (a) it's easier to code and (b) while you have defined a maximum concurrent attachment limit, you may sometimes want to attach less than the maximum number of files.

## Attaching files stored on your web server

So far, we've covered uploading attachments using your web browser. You may also want to attach files that are already on your web server. (For example, perhaps you want to attach a report that is automatically created by another system and saved to a location on the web server).

To enable filesystem attachments, look for the following line in `config/config.php`:

```
define("FILESYSTEM_ATTACHMENTS", 0);
```

Change `FILESYSTEM_ATTACHMENTS` from 0 to 1 to enable filesystem attachments.

Now, on the **send a message** page, under the **Attach** tab, in addition to a **Browse** button to upload the file, you also have an input field allowing you to specify a filesystem location for this file:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

---

Content
Format
Attach
Scheduling
Criteria
Lists
Misc

**? Add attachments to your message**

The upload has the following limits set by the server:  
Maximum size of total data being sent to the server: 8M  
Maximum size of each individual file: 10M

New Attachment  Browse... Add (and save)

or Path to file on server:

Description of attachment:

---

Send Test Message to email address(es):   
(comma separated addresses - all must be users)

---

Save Changes

- users
- manage users
- subscribe pages
- messages
- statistics

---

- templates
- process queue
- manage bounces
- eventlog

Note that no validation is done at this point. That is, phpList doesn't stop you if the file doesn't exist. Your only indication that this file may not exist is the file size (zero in the following example):

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

Information: Added attachment 1 mime: image/png

**Message saved**

**Content** **Format** **Attach** **Scheduling** **Criteria** **Lists** **Misc**

? Add attachments to your message

The upload has the following limits set by the server:  
Maximum size of total data being sent to the server: 8M  
Maximum size of each individual file: 10M

| Current     | filename                          | desc                                       | size | file | del                      |
|-------------|-----------------------------------|--|------|------|--------------------------|
| Attachments |                                   |  |      |      |                          |
| 7           | /tmp/my-super-cool-attachment.png | Something I uploaded to the server earlier | 0    | X    | <input type="checkbox"/> |

DELETE CHECKED

close ^

New Attachment

or Path to file on server:

Description of attachment:

|                 |
|-----------------|
| users           |
| manage users    |
| subscribe pages |
| messages        |
| statistics      |
| templates       |
| process queue   |
| manage bounces  |
| eventlog        |

If the file size is a positive number, then phpList is able to find the file, and therefore attach it (assuming that the web server user has permission to read that file on the filesystem).

Note that phpList will not warn you of a failure to attach a filesystem file – it will silently ignore it. That is, in the following example, only the second attachment (testattachment.txt) will actually be sent, but no errors will be produced:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

Information: Added attachment 1 mime: text/plain

**Message saved**

Content Format **Attach** Scheduling Criteria Lists Misc

? Add attachments to your message

The upload has the following limits set by the server:  
 Maximum size of total data being sent to the server: 8M  
 Maximum size of each individual file: 10M

| Current            | filename                          | desc                      | size | file | del                      |
|--------------------|-----------------------------------|---------------------------|------|------|--------------------------|
| <b>Attachments</b> |                                   |                           |      |      |                          |
| 9                  | /tmp/my-super-cool-attachment.png | This doesn't really exist | 0    | ✗    | <input type="checkbox"/> |
| 10                 | /tmp/testattachment.txt           | This does                 | 6    | ✗    | <input type="checkbox"/> |

DELETE CHECKED

close ▲

New Attachment

or Path to file on server:

Description of attachment:

users  
 manage users  
 subscribe pages  
 messages  
 statistics  
 templates  
 process queue  
 manage bounces  
 eventlog

However, when examining the message after sending using the **Edit Message** link, the **Attach** tab will show which attachments were successfully attached:

Content Format **Attach** Scheduling Criteria Lists Misc

? Add attachments to your message

The upload has the following limits set by the server:  
 Maximum size of total data being sent to the server: 8M  
 Maximum size of each individual file: 10M

| Current            | filename                          | desc                      | size | file | del                      |
|--------------------|-----------------------------------|---------------------------|------|------|--------------------------|
| <b>Attachments</b> |                                   |                           |      |      |                          |
| 9                  | /tmp/my-super-cool-attachment.png | This doesn't really exist | 0    | ✗    | <input type="checkbox"/> |
| 10                 | /tmp/testattachment.txt           | This does                 | 6    | ✓    | <input type="checkbox"/> |

DELETE CHECKED

templates  
 process queue  
 manage bounces  
 eventlog

## Securing your attachments stored on the web server

By default, any attachments you upload are stored in `/tmp` on your web server. This is not ideal, as on a shared host, other users might also have access to these files.

It's recommended that you create a dedicated directory for your attachments instead. It may be best to create an "attachments" folder at the same level as the folder which contains your publicly accessible HTML and PHP files (but not in it!), ensuring that this directory is available to phpList but not to normal web users.



Make sure that the web server has "write" access to this directory, or else it won't be able to save the attachments you upload.

When you've created this directory, open `config/config.php` and look for this line:

```
$attachment_repository = '/tmp';
```

Change the value of `$attachment_repository` to refer to the full path to the new "attachments" directory you created, for example: `/home/myuser/mywebsite/attachments`.

## Automatically repeat messages

You may need to have messages automatically repeat every day, week, or month. For example, perhaps you have a filesystem attachment (a daily report) that you want e-mailed to your subscribers at 7 AM every morning.

To enable message repetition, look for this line in `config/config.php`:

```
define("USE_REPETITION", 0);
```

Change `USE_REPETITION` from 0 to 1 to enable message repetition.

On the **send a message** page, under the **Scheduling** tab, you now have the option for hourly, daily, or weekly repetition. You can also schedule a date when the repetition will cease:

To send a message fill out the information below. The minimum requirement for a valid message is the information in the "Content" tab. Make sure to save your changes before going to a new tab or they will be lost. To finish your message and send it for real, choose the "Lists" tab, select your lists and click the button "Send Message to Selected Mailinglists"

users  
manage users  
subscribe pages  
messages  
statistics

templates  
process queue  
manage bounces  
eventlog

**Content** Format Attach Scheduling Criteria Lists Misc

? Embargoed Until: 18 November 2010 21 00

? Repeat message every: Day

Repeat Until: 18 December 2010 21 00

Send Test Message to email address(es):   
(comma separated addresses - all must be users)

Save Changes

## Sending a repeating message

Aside from the **Scheduling** tab, sending a repeating message is just like sending a normal message. As soon as the message has finished sending, another copy of that message will be put in the **queued** tab to be sent when the hourly / daily / weekly embargo expires:

If you view a message, you will be able to resend it to a different list

sent draft **queued**

1 Messages

Listing message 1 to 1

<< < > >>

| Message info   | Status          | Action  |
|--|-----------------|---------|
| From: Webmaster noreply@phplist.demo.funkypenguin.info | submitted       | view    |
| Subject: test repeating message                        |                 | requeue |
| Entered: 2010-11-18 22:00:52                           | suspend sending | edit    |
| Embargo: 2010-11-19 21:00:00                           |                 | delete  |

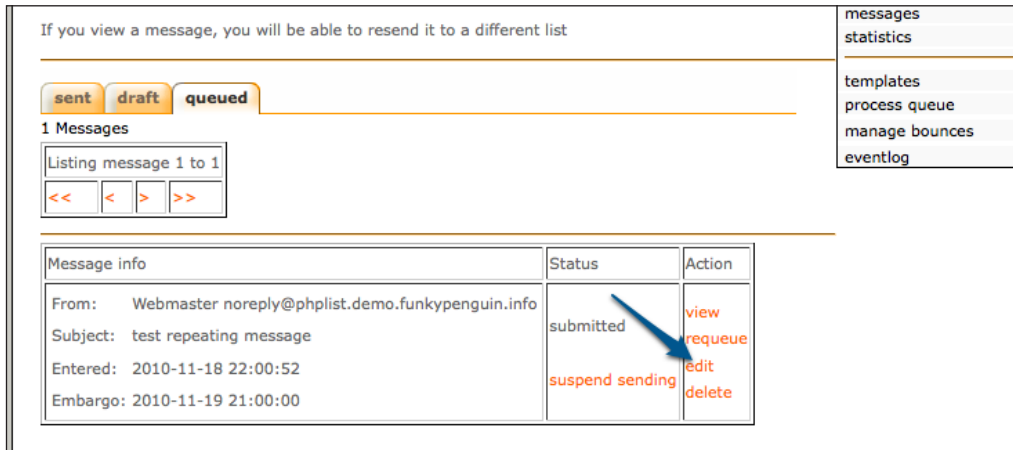
messages  
statistics

templates  
process queue  
manage bounces  
eventlog

## Forcing a repeating message to repeat before embargo

Under certain circumstances, you may want to force a repeating message to be resent before its embargo expires.

To do so, view the **Queued** tab and click on **edit** to edit the message:



If you view a message, you will be able to resend it to a different list

sent draft **queued**

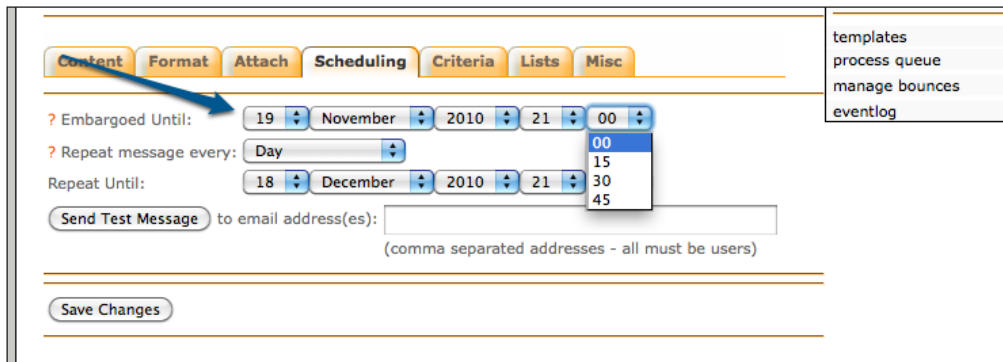
1 Messages

Listing message 1 to 1

| Message info  | Status    | Action                            |
|---|-----------|-----------------------------------|
| From: Webmaster noreply@phplist.demo.funkypenguin.info<br>Subject: test repeating message<br>Entered: 2010-11-18 22:00:52<br>Embargo: 2010-11-19 21:00:00 | submitted | view<br>requeue<br>edit<br>delete |

messages  
statistics  
templates  
process queue  
manage bounces  
eventlog

Click on the **Scheduling** tab and change the **Embargoed Until** fields to the new time (or a time that has already passed), and save the message:



Content Format Attach **Scheduling** Criteria Lists Misc

? Embargoed Until: 19 November 2010 21 00

? Repeat message every: Day

Repeat Until: 18 December 2010 21 30

Send Test Message to email address(es):  
(comma separated addresses - all must be users)

Save Changes

templates  
process queue  
manage bounces  
eventlog

The message will be sent on the next **process queue** action and unless you also changed the **Repeat Until** fields, the next repetition of this message will be sent at the normal scheduled time.

## Auto-generating messages from RSS feeds

One of phpList's unique strengths is its ability to process RSS feeds and insert them into your messages. For example, you may distribute a formal daily newsletter, but include your latest twitter posts at the bottom.

There are some requirements to processing RSS feeds with phpList:

- Your PHP installation must have XML support. If you don't have this, or don't know, ask your web hosting provider.
- Your RSS feed must be accessible (via HTTP) to your phpList installation.
- Your PHP instance should be running with `safe_mode` off. Some web hosting providers will enable `safe_mode` for security reasons—unfortunately, this can block the RSS-fetching operation of phpList.
- You must have enabled phpList's repeating messages feature.



You can test whether your PHP includes XML support and determine the status of `safe_mode` by creating a file called `test.php`, which contains only the following line:

```
<?php phpinfo(); ?>
```

Then upload this file to your hosting provider and open its public URL in your web browser. PHP will give you detailed information about your installation.

## Enabling RSS support

To enable RSS support, look for the following line in `config/config.php`:

```
define("ENABLE_RSS", 0);
```

Change `ENABLE_RSS` from 0 to 1 to enable RSS support.

If you have successfully enabled RSS support, you will see three new RSS-related functions on the main admin page (**getrss**, **viewrss**, and **purgerss**):

|                                |                           |         |
|--------------------------------|---------------------------|---------|
| <b>Message functions</b>       |                           | close ✕ |
| <a href="#">send</a>           | Send a message            |         |
| <a href="#">templates</a>      | Configure Templates       |         |
| <b>Messages</b>                | List all Messages         |         |
| <a href="#">processqueue</a>   | Process the Message queue |         |
| <a href="#">processbounces</a> | Process Bounces           |         |
| <a href="#">Bounces</a>        | View Bounces              |         |
|                                |                           | close ⬆ |
| <b>RSS Functions</b>           |                           |         |
| <a href="#">getrss</a>         | Get RSS feeds             |         |
| <a href="#">viewrss</a>        | View RSS items            |         |
| <a href="#">purgerss</a>       | Purge RSS items           |         |
|                                |                           | close ⬆ |



## Associating an RSS feed with a list

phpList allows you to associate a single RSS feed with a single list. To add a feed to a list, edit the list, and add the RSS feed URL to the **RSS Source** input field:

|  |   |                 |
|--|---|-----------------|
| <b>members of this list</b>                      |   | send a message  |
| List name:                                       | <input type="text" value="myfans"/>   | users           |
| Check this box to make this list active (listed) | <input checked="" type="checkbox"/>   | manage users    |
| Order for listing                                | <input type="text" value="0"/>  | subscribe pages |
| Owner  | <input type="text" value="admin"/>  | messages        |
| RSS Source                                       | <input type="text" value="http://twitter.com/statuses/user_timeline/17778401.rss"/> | statistics      |
| List Description                                 | <input type="text" value="Fans of my work. My peeps."/>                             | templates       |
|  |   | process queue   |
|  |   | manage bounces  |
|  |   | eventlog        |



### Validating a feed

You can confirm that the RSS feed URL is valid by running it through the "validator" at <http://validator.w3.org/feed/>.

When you view the summary of your lists, you'll now see the RSS feed associated with each list:

| No  | Name          | Order                          | Functions  | Active                              | Owner |
|---|---------------|--------------------------------|--|-------------------------------------|-------|
| 1   | <b>myfans</b> | <input type="text" value="0"/> | <a href="#">edit</a>   <a href="#">view members</a>   <a href="#">delete</a> | <input checked="" type="checkbox"/> | admin |
| 1 Members   |               |                                |  |                                     |       |
| RSS source: <a href="http://twitter.com/statuses/user_timeline/17778401.rss">http:// twitter.com/ statuses/ user_timeline/ 17778401.rss</a> |               |                                |  |                                     |       |
| <a href="#">(view items)</a>  |               |                                |  |                                     |       |
| Fans of my work. My peeps.  |               |                                |  |                                     |       |

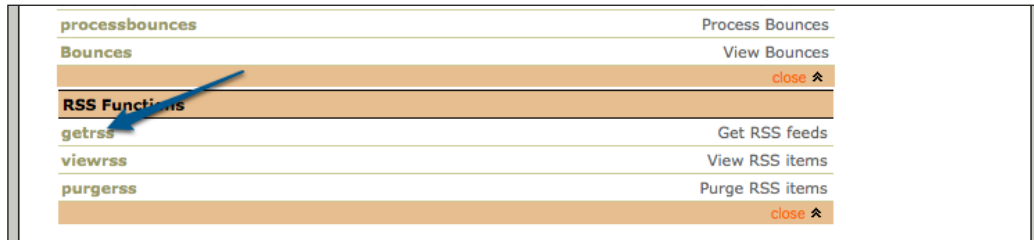
However, you'll notice that if you click on **(view items)**, there will be no items displayed:

|                      |                 |
|----------------------|-----------------|
| RSS items for myfans | send a message  |
| 0 RSS Items          | users           |
|                      | manage users    |
|                      | subscribe pages |
|                      | messages        |

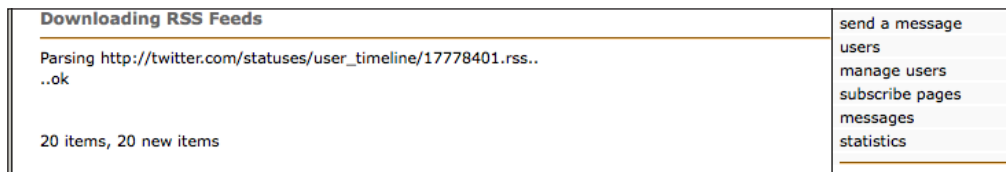
This is because, like the message queue, each RSS feed must be periodically "processed" for new items, before phpList will know about them.

## "Getting" new RSS items

To "get" new RSS items, click on the new `getrss` link at the bottom of the main admin page:



phpList will now connect to the URL and use the built-in XML libraries to process the RSS feed and put each item into the database. This `getrss` action is the acid test of your RSS implementation – if this doesn't work, then address the requirements with your web hosting provider or test your RSS feed for validity, before proceeding with sending an actual message.



## User requirements to receive RSS messages

Because RSS messages are designed to take advantage of the "repeating messages" function of phpList, in order for users to receive these messages, they must have already selected an RSS frequency of either **hourly**, **daily**, or **weekly**.

You can confirm a user's RSS frequency setting by examining their user account:

|  |                                   |                 |
|--|-----------------------------------|-----------------|
| Delete will delete user and all listmemberships  |                                   | subscribe pages |
|  |                                   | messages        |
|  |                                   | statistics      |
| herbertneedleman@phplistpackt.com <a href="#">update page</a> <a href="#">unsubscribe page</a> <a href="#">history</a> |                                   | templates       |
| <b>User Details</b>  |                                   | process queue   |
|  |                                   | manage bounces  |
|  |                                   | eventlog        |
| ID   | 1                                 |                 |
| Email  | herbertneedleman@phplistpackt.com |                 |
| Is this user confirmed (1/0)   | 1                                 |                 |
| Is this user blacklisted   | 0                                 |                 |
| Number of bounces on this user   | 0                                 |                 |
| Entered  | 2010-11-18 08:05:28               |                 |
| Last Modified  | 2010-11-18 23:09:21               |                 |
| Unique ID for User   | dfd5991a689f9f9eb2ade9fceab0a698  |                 |
| Send this user HTML emails   | 1                                 |                 |
| Which page was used to subscribe   |                                   |                 |
| RSS Frequency  | daily                             |                 |
| Password   |                                   |                 |
| Last time password was changed   | 2010-11-18                        |                 |
| Is this account disabled?  | 0                                 |                 |
| Additional data  |                                   |                 |
| Foreign Key  |                                   |                 |
| <input type="button" value="Save Changes"/>  |                                   |                 |
| <b>Mailinglist Membership:</b>   |                                   |                 |
| <input checked="" type="checkbox"/> myfans   |                                   |                 |
| <input type="button" value="Save Changes"/>  |                                   |                 |

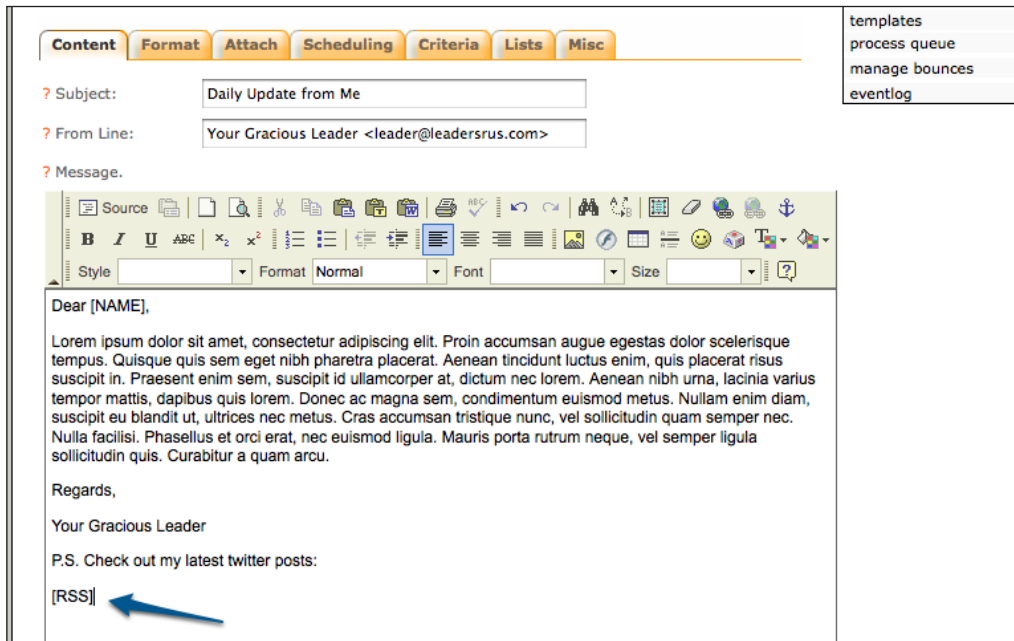


If you plan to make use of repeating RSS messages, make sure that your subscribe form includes the option to select a frequency. If you want to force any new subscribers into a particular frequency (that is, they are deliberately subscribing for RSS updates), you can simply restrict the RSS frequency options to a single option (such as **daily**).

## Sending a message including RSS

Now that you have a working RSS feed, it's time to send a message that will include the items in this feed. As usual, use the **send a message** link to compose a new message to the appropriate list.

Ensure you enter the placeholder **[RSS]** somewhere in your message:



## Setting your RSS message's schedule

If you simply send a message containing **[RSS]** as a standard, one-off message, the RSS feed will not be inserted into the message. For a message to include RSS items, phpList needs to know whether this message is destined for users who've selected an hourly, daily, or weekly RSS frequency.

On the **Scheduling** tab, select your RSS frequency. Note that when you click on the **Save** button, the **Repeat message every** drop-down box will automatically be adjusted to match your RSS frequency selection. That is, repeating messages is a pre-requisite of RSS support.

Content Format Attach **Scheduling** Criteria Lists Misc

manage bounces  
eventlog

? Embargoed Until: 18 November 2010 23 00

? Repeat message every: -- no repetition

Repeat Until: 18 November 2011 00 00

If you want to use this message as the template for sending RSS feeds select the frequency it should be used for and use [RSS] in your message to indicate where the list of items needs to go.

none  Daily  Weekly  Monthly

Send Test Message to email address(es):

(comma separated addresses - all must be users)

Once you've saved the message, the next time the queue is processed, any new RSS items will be included in the message to each user:

enim, quis placerat risus suscipit in. Praesent enim sem, suscipit id ullamcorper at, dictum nec lorem. Aenean nibh urna, lacinia varius tempor mattis, dapibus quis lorem. Donec ac magna sem, condimentum euismod metus. Nullam enim diam, suscipit eu blandit ut, ultrices nec metus. Cras accumsan tristique nunc, vel sollicitudin quam semper nec. Nulla facilisi. Phasellus et orci erat, nec euismod ligula. Mauris porta rutrum neque, vel semper ligula sollicitudin quis. Curabitur a quam arcu.

Regards,  
Your Gracious Leader

P.S. Check out my latest twitter posts:

**myfans**

packtpub: Special offer on range of Packt IBM books. Buy any IBM book and get upto 40% off. <http://bit.ly/cBD6V>

packtpub: Special offer on range of Packt IBM books. Buy any IBM book and get upto 40% off. <http://bit.ly/cBD6V>

packtpub: Results in! Open Source E-Commerce Applications category: Winner @PrestaShop 2nd Place @OpenCart 3rd Place @TomatoCart <http://bit.ly/c24fXi>

packtpub: Results in! Open Source E-Commerce Applications category: Winner @PrestaShop 2nd Place @OpenCart 3rd Place @TomatoCart <http://bit.ly/c24fXi>

packtpub: Congratulations to @PrestaShop, winner of the Open Source E-Commerce Applications category in the 2010 Open Source Awards #OpenSource

packtpub: Congratulations to @PrestaShop, winner of the Open Source E-Commerce Applications category in the 2010 Open Source Awards #OpenSource

## Setting up processing automation

Now that we are able to manually send messages and process bounces, it would be handy to be able to automate these actions so that they happen at regular intervals.

---

## Requirements

The requirements are as follows:

- Access to Cron – for automatic phpList processing, we'll need access to schedule jobs on our server using cron entries. Your provider may have made this available using a control panel and it may be called something like "scheduled tasks", or "automated tasks".
- Filesystem information – we'll also need to know the filesystem path to our phpList installation, the user our scripts will run as and (potentially) the path to the PHP binary. It will also be helpful to have shell (SSH) access to our host.

## Customize the CLI "wrapper"

Before we can use the scheduling tools to set up automated phpList processing, we need to customize the "wrapper" bash shell which we can run on our host, that'll do the CLI-based phpList work.

phpList's source code includes a `bin` subfolder containing a file called `phplist` (this file will be set as executable if you unpacked the phpList source on a POSIX-based host).

Copy the `bin` directory to a location outside of your publicly-accessible phpList files (that is, at the same level in the directory hierarchy as your `public_html` folder or the folder where all your web-accessible files are).

Edit the `phplist` file inside the `bin` directory.

You will need to customize the following lines to suit your installation:

```
CONFIG=/home/website/public_html/lists/config/config.php
```

Change this to reflect the filesystem path to your phpList installation:

```
/usr/bin/php /home/website/public_html/lists/admin/index.php $*
```

Change this to reflect both the filesystem path (again) and the PHP binary location on our host (`/usr/bin/php` may well be correct).

Having updated the `bin/phplist` file, open your `config/config.php` file and look for the line containing `commandline_users`:

```
$commandline_users = array("admin");
```

Edit this line by replacing `admin` with the user under which your web host processes run as.



This username might be difficult to determine, especially in a shared environment. If you can access a shell session, run the `whoami` command to determine this. If you are using a dedicated server, this will be the user under which your web server runs. Your web hosting provider can supply you with this information.

Now that you have prepared phpList's config and your wrapper script, use whatever interface you have at your disposal to schedule tasks as follows:

**Process bounces:** `/path/to/bin/phplist -p processbounces`

**Process queue :** `/path/to/bin/phplist -p processqueue`

**Process RSS :** `/path/to/bin/phplist -p processrss`



Don't worry about running the `processqueue` task too frequently – if it runs and detects that it's been run in the past six minutes, it'll abort with a warning so you can't cause multiple copies to run simultaneously.

The author recommends you to run the `processqueue` action every 30 minutes and the `processbounces` and `processrss` every day.



#### **Getting RSS before processing queue**

If you have a regular, repeating RSS message scheduled for sending in your queue, it's a good idea to make sure that the `processrss` task runs before this message is scheduled to be sent. That way, the latest RSS items will be included in the message.

## Summary

In this chapter, you've learned to take advantage of some of phpList's more advanced features, such as:

- Advanced bounce handling
- Domain-based e-mail throttling
- Adding attachments to messages
- Automatically repeating messages
- Including RSS feeds in messages
- Setting up processing automation

In the following chapter, you'll learn how to add your own advanced features by modifying (hacking) the phpList code.





# 10

## Hacking phpList

In this chapter, you will learn how to add extra functionality to your phpList installation by "hacking" the source code. Unlike the advanced features previously covered, none of the hacks covered in this chapter are officially supported by phpList and may not work in future versions. Instead of teaching you how to implement specific features, this chapter is aimed more at teaching the basic methods behind adding functionality to phpList by either making direct code changes or implementing pre-packaged, community-developed hacks.

We will cover the following hacks:

- Substituting user attributes in the message subject line
- Sending system messages formatted as HTML instead of plain text
- Creating a messages archive page
- Sending messages through your e-mail client

### **Substituting user attributes in the subject line**

phpList supports the substituting of attributes into your messages recorded from your subscribers by using [ATTRIBUTE] tags in the content of the message.

It doesn't, however, support this in message subjects. A personalized e-mail subject line goes a long way towards engaging your readers, so let's modify phpList to perform the same attribute substitution in the subject line.

Open `admin/sendemaillib.php` and look for the following stanza, around line #757:

```
if (!TEST) {
    if ($hash != 'forwarded' || !sizeof($forwardedby)) {
        $fromname = $cached[$messageid] ["fromname"];
        $fromemail = $cached[$messageid] ["fromemail"];
        $subject = $cached[$messageid] ["subject"];
    } else {
        $fromname = '';
        $fromemail = $forwardedby['email'];
        $subject = $GLOBALS['strFwd'].': '.$cached[$messageid] ["subject"];
    }
};
```

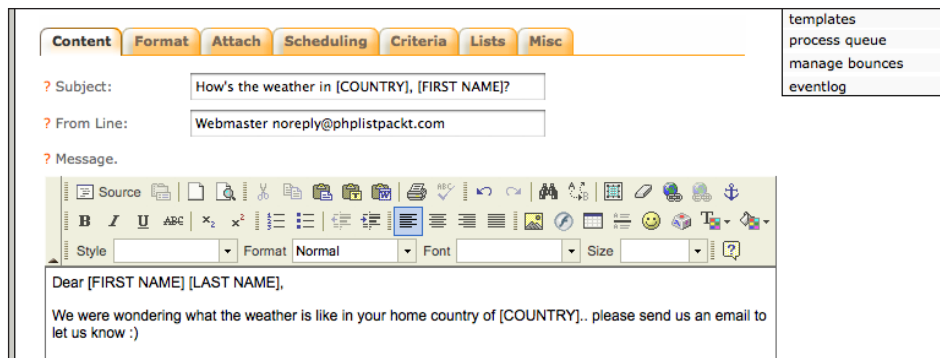
This is the logic that decides whether an e-mail is being sent to a subscriber or being forwarded to a third party by a subscriber. Instead of interfering with this code, we'll do our modifications directly afterwards.

Directly after this stanza, paste in the following code:

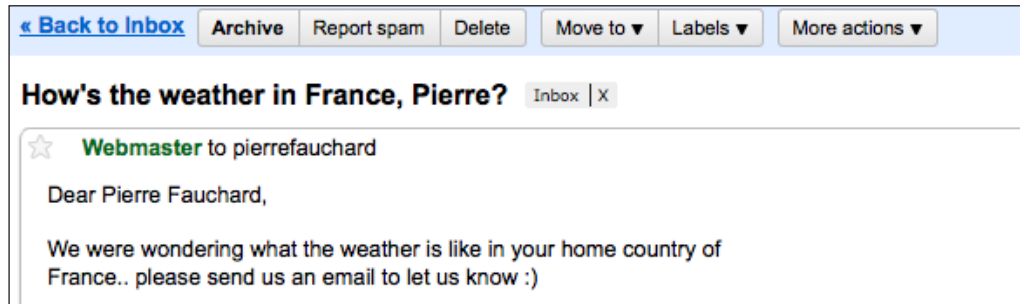
```
// START HACK - Perform attribute substitution in the subject
if (is_array($user_att_values)) {
    foreach ($user_att_values as $att_name => $att_value) {
        $subject = str_ireplace("[$att_name]", $att_value,
    $subject);
    }
}
// END HACK - Perform attribute substitution in the subject
```

This cycles through the in-scope variable—`$user_att_values` and performs case-insensitive attribute replacement against the `$subject` variable.

Now we can attempt to send a message with a personalized subject line:

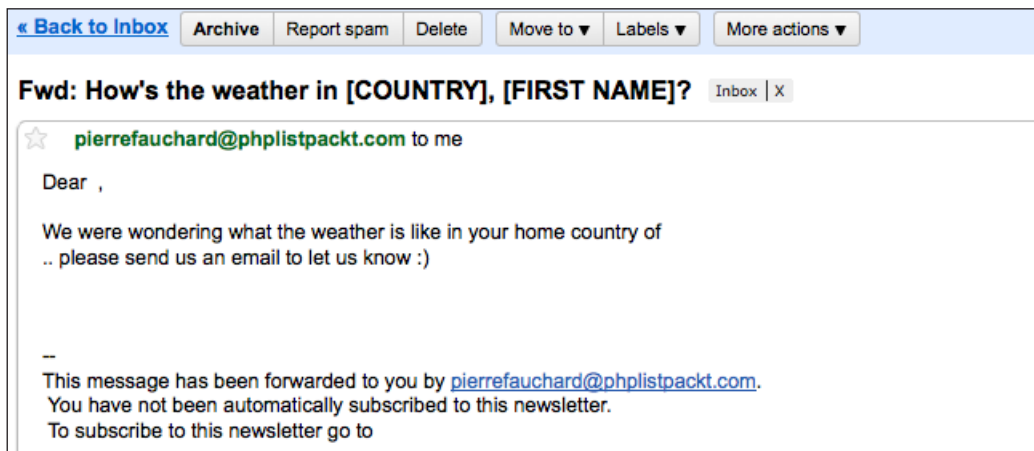


The messages are received and individually customized for our subscribers:



## Caveat #1 – no attribute substitution for a third party

You'll notice in the original stanza before our pasted hack that it's also possible for a subscriber to click a link to forward a message that they received onto a third party. This will become a problem for our attribute-substitution code, as we don't have any attributes on record for the third party. The third party will receive a message without any attribute substitution:



## Caveat #2 – "Forward message" page displays an un-substituted subject

The mechanism for this forwarding process involves our subscriber clicking on a link in the message to forward the message. This page displays the subject of the message (un-substituted) when requesting the e-mail address to forward the message:

PHPLIST

powered by:  
[PHP] + [MySQL]

PHPLIST

Forwarding the message with subject How's the weather in [COUNTRY], [FIRST NAME]?

Please enter a valid email address to forward to

Continue

POWERED BY  
PHPLIST

© tincan limited | phplist - version

## Sending system messages as HTML instead of text

While phpList allows you to send beautifully formatted HTML messages, its own system messages (confirmation request, unsubscription notification, and so on) are sent in plain-old-boring-text.

This hack allows you to enter HTML in your system messages and have them delivered in multipart HTML-and-text.

## Code changes

The following pages detail all the code changes required for this hack. A patch file is included with this book, as well as a readme file that explains how to apply it.

The following pages detail the code changes that are required, step-by-step.

## Creating the plain-text part of the message by stripping out the HTML

You'll customize your system message to include HTML code. Unlike the message composition interface, phpList doesn't give us the option to define separate HTML and text parts to our system messages, so we'll have to create the text version by stripping out all of the HTML code, but still preserving any hyperlinks.

Open `admin/lib.php` and look for the following line (around line #271):

```
$destinationemail = '';
```

Directly below this line, add the following:

```
// START HACK - Allow HTML in system messages
$html_message = $message;
$message = preg_replace('/(<script.*\/script>)|(<style.*\/style>)/i', '', $message);
$message = preg_replace('/<a\s*. *?href\s*=\s*(\"|\')([^\"]*\')(\"|\'). *?>(.*?)<\/a>/i', "$4:\n$2\n", $message);
$message = strip_tags($message);
// END HACK - Allow HTML in system messages
```

This code starts off with the assumption that the message that it gets from the database includes all the HTML code required. As you'll also want to generate a plain-text version of this message (for multipart MIME messages) it copies the HTML message into a new variable for the plain text message.

Then, it removes any JavaScript or CSS styles which may contain URLs. Next, it converts any HTML links into plain text links and then it strips out all remaining HTML.

## Sending both the HTML and text versions of the message

Now that we have two separate messages – one HTML and one plain-text – we'll modify the code that actually sends the message ensuring that it sends both parts.

Open `admin/lib.php` and look for the following lines (around line #283):

```
if (!ereg("dev",VERSION)) {
    $mail = new PHPListMailer('systemmessage',$to);
    $destinationemail = $to;
    $mail->add_text($message);
} else {
    # send mails to one place when running a test version
```

```
$message = "To: $to\n".$message;
if ($GLOBALS["developer_email"]) {
    # fake occasional failure
    if (mt_rand(0,50) == 1) {
        return 0;
    } else {
        $mail = new PHPListMailer('systemmessage', $GLOBALS["developer_email"]);
        $mail->add_text($message);
        $destinationemail = $GLOBALS["developer_email"];
    }
} else {
    print "Error: Running CVS version, but developer_email not set";
}
}
```

This conditional implements slightly different message-sending behavior, depending on whether phpList's version indicates that it's a development instance. Our addition is the same in both cases, so identify both occurrences of the following line in this stanza:

```
$mail->add_text($message);
```

And below each instance, add the following:

```
$mail->add_html($html_message); // HACK - Allow HTML in system messages
```

This simply uses already-existing features of the `PHPListMailer` class to add an HTML part to the message.

## Stripping slashes from messages stored in the database

phpList stores the custom system messages in the database in a "quoted" format. That is, any apostrophes will be preceded with backslashes. This is a common (and sensible) security practice for protection against SQL injection attacks. However, because phpList doesn't anticipate that these custom messages will contain HTML, it doesn't strip these slashes off the content before sending them. We'll have to manually do that in the following section, so that our HTML messages aren't sent full of slashes.

---

## Stripping slashes from a subscribe message for normal subscriptions

Open `admin/subscribe/lib2.php` and look for the following line (around line #312):

```
$subscribemessage = ereg_replace('\[LISTS\]', $lists, getUserConfig("subscribemessage:$id", $userid));
```

Directly below this line, add the following:

```
$subscribemessage = stripslashes($subscribemessage); // HACK - Allow HTML in system messages
```

## Stripping slashes from member data change notifications

Open `admin/subscribe/lib2.php` and look for the following line (around line #595):

```
$message = ereg_replace('\[USERDATA\]', $datachange, $message);
```

Directly below this line, add the following:

```
$message = stripslashes($message); // HACK - Allow HTML in system messages
```

## Stripping slashes from the confirmation message

Open `index.php` and look for the following line (around line #555):

```
$confirmationmessage = ereg_replace('\[LISTS\]', $lists, getUserConfig("confirmationmessage:$id", $userdata["id"]));
```

Directly below this line, add the following:

```
$confirmationmessage = stripslashes($confirmationmessage); // HACK - Allow HTML in system messages
```

## Stripping slashes from the unsubscribe message

Open `index.php` and look for the following line (around line #656):

```
$unsubscribemessage = ereg_replace("\[LISTS\]", $lists, getUserConfig("unsubscribemessage", $userid));
```

Directly below this line, add the following:

```
$unsubscribemessage = stripslashes($unsubscribemessage); // HACK - Allow HTML in system messages
```



## Stripping slashes from the personal location message

Open `index.php` and look for the following line (around line #130):

```
sendMail ($uid[1],getConfig("personallocation_subject"),getUserConfig
("personallocation_message",$uid[2]),system_messageheaders(),$GLOBALS
["envelop
e"]);
```

Replace this line with the following:

```
// START HACK - Allow HTML in system messages
$personallocationmessage = getUserConfig("personallocation_
message",$uid[2]);
$personallocationmessage = stripslashes($personallocationmessage);
sendMail ($uid[1],getConfig("personallocation_subject"),$persona
llocationmessage,system_messageheaders(),$GLOBALS["envelope"]);
// END HACK - Allow HTML in system messages
```

## Stripping slashes from the subscribe message for user imports

Open `admin/import1.php` and look for the following line (around line #246):

```
$subscribemessage = ereg_replace('\[LISTS\]', $listoflists, getUserCon
fig("subscribemessage",$userid));
```

Directly below this line, add the following:

```
$subscribemessage = stripslashes($subscribemessage); // HACK - Allow
HTML in system messages
```

Open `admin/import3.php` and look for the following line (around line #461):

```
$subscribemessage = ereg_replace('\[LISTS\]', $listoflists, getUserCon
fig("subscribemessage",$userid));
```

Directly below this line, add the following:

```
$subscribemessage = stripslashes($subscribemessage); // HACK - Allow
HTML in system messages
```

Open `admin/commonlib/pages/importcsv.php` and look for the following line (around line #834):

```
$subscribemessage = ereg_replace('\[LISTS\]', $listoflists, getUserCon
fig("subscribemessage", $userid));
```

Directly below this line, add the following:

```
$subscribemessage = stripslashes($subscribemessage); // HACK - Allow
HTML in system messages
```

## Stripping slashes from the subscribe message when resending confirmations

Open `admin/reconcileusers.php` and look for the following stanza (around line #36):

```
if ($userdata["subscribepage"]) {
    $subscribemessage = ereg_replace('\[LISTS\]', $lists, getUserConfig("subscribemessage:".$userdata["subscribepage"],$id));
    $subject = getConfig("subscribesubject:".$userdata["subscribepage"]);
} else {
    $subscribemessage = ereg_replace('\[LISTS\]', $lists, getUserConfig("subscribemessage",$id));
    $subject = getConfig("subscribesubject");
}
```

Directly below this stanza, add the following:

```
$subscribemessage = stripslashes($subscribemessage); // HACK - Allow
HTML in system messages
```

## Web interface changes

Having made all the required code changes, we'll now customize the contents of the system messages in the web interface.

## Standard system messages

The standard system messages are configured under the **Configure** page on the web interface.

You should now modify the following:

- Message users receive when they subscribe
- Message users receive when they unsubscribe
- Message users receive after confirming their e-mail address
- Message that is sent when users change their information
- Part of the message that is sent to their new e-mail address when users change their information and the e-mail address has changed



## Subscription and confirmation messages on the pre-existing subscribe page

While you've made changes to the subscribe and confirmation messages on the **Configure** page, be aware that these changes only apply to any subsequently created subscribe pages.

If you already have subscribe pages set up, you'll also need to customize the subscribe and confirmation messages which were inherited by these pages on creation:

Display Email confirmation  Display email confirmation  Don't display email confirmation

**Message they receive when they subscribe**

Subject: Request for confirmation

Message: Almost welcome to our newsletter(s) ...  
Someone, hopefully you, has subscribed your email address to the following newsletters:  
[LISTS]  
If this is correct, please click the following link to confirm your subscription.  
Without this confirmation, you will not receive any newsletters.

**Message they receive when they confirm their subscription**

Subject: Welcome to our Newsletter

Message: `<p>&nbsp;&nbsp;&nbsp; Welcome to our <span style="color: rgb(51, 153, 102);"><strong>Newsletter</strong></span><br /><br />&nbsp;&nbsp;&nbsp; (<em>Please keep this email for later reference</em>)<br /><br />&nbsp;&nbsp;&nbsp; Your email address has been added to the following newsletter(s):<br />&nbsp;&nbsp;&nbsp; [LISTS]<br /><br />&nbsp;&nbsp;&nbsp; To update your details and preferences please go to`

### Tips

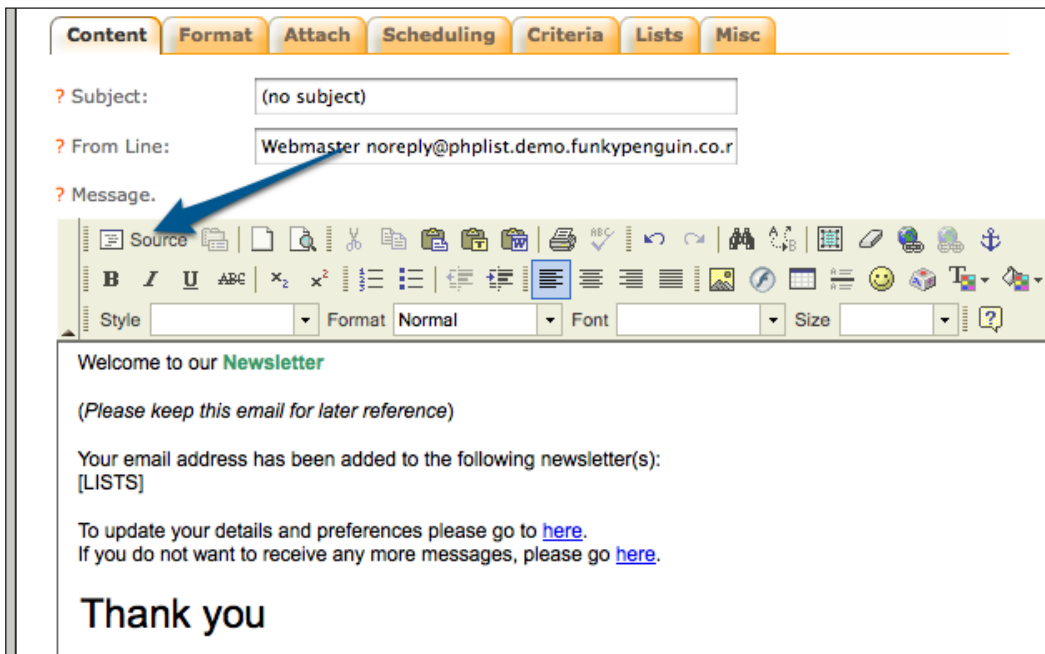
The following are a few tips to help you avoid common stumbling blocks in sending system messages by HTML.

### Be sure to customize all system messages

Once this hack is enabled, all system messages will be sent as HTML. If a default phpList message is sent as HTML, it will lose all its line breaks and hence look worse than the original. At the very least, you should convert word-for-word, each plain text message into HTML.

## Composing HTML messages

A useful way to convert a plain text system message to HTML is to copy the text contents and paste it into a new message. Then customize the message using the WYSIWYG editor and click the **Source** button to view the generated HTML code:



## How to deal with placeholders producing URLs

You'll note that some messages include placeholders like [SUBSCRIBEURL], which will be converted into a user-specific URL at the time of sending. Because you want these links to still be clickable when sent via HTML, be sure to check that the <A HREF> link contains the correct source.

For example, if the text version of a message looks like this:

```
If this is correct, please click the following link to confirm your
subscription.
```

```
Without this confirmation, you will not receive any newsletters.
```

```
[CONFIRMATIONURL]
```

---

In this example, the corresponding HTML version should look like this:

```
<p>If this is correct, please click the following link to confirm
your subscription.</p>
  <p>Without this confirmation, you will not receive any
newsletters.</p>

  <p><A HREF="[CONFIRMATIONURL]">Click here</A></p>
```

## Keep CSS styling inline

This tip applies not just to system messages as HTML, but also to any HTML messages sent through phpList.

Many online e-mail providers (for example, Gmail) will attempt to strip out any potentially harmful or incompatible style information from your message before displaying it. If you (as you would on a regular website) apply all your styling before the `<body>` tag, it's likely it'll all be stripped out.

A more reliable, but more time-consuming, approach is to apply all your styling inline.

Say you want a particular paragraph to have a gray background with a 1 px solid border on all sides. Instead of declaring a custom style `greyborderbox` in the `<head>` section of the HTML, and then applying that to your paragraph, you should present your paragraph like this:

```
<p style="border:1px solid; background: grey;">Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Vivamus pretium, dolor in eleifend
tempor, lacus erat ultricies sapien, in ultrices lorem ligula quis
eros. Etiam at velit convallis nulla tristique elementum quis at
lacus. Pellentesque ultricies facilisis justo eu egestas.</p>
```

This will ensure maximum compatibility with web mail providers.

## Creating a messages archive page

While the previous section explained how to add functionality to phpList by manual editing of the source code, this section will walk you through implementing a "ready-built" phpList hack – the messages archive page.

Perhaps you want your readers to be able to refer to an "archives" page to be able to find and re-read your previous mailings. phpList itself does not provide such an archive, but some committed users have implemented an effective solution.

The solution is discussed in the phpList forums at <http://forums.phplist.com/viewtopic.php?t=1501> and the necessary code is available at <http://docs.phplist.com/NewsletterArchive>. Copy the contents of the wiki page (everything from the opening `<?php` tag at the top to the closing `?>` tag at the bottom), and paste it into a new file called `archive.php` at the root of your publicly accessible phpList files.

You'll recall from *Chapter 8, Securing phpList*, that the `.htaccess` file in the root of your phpList installation controls which files may be viewed by web users.

Open `.htaccess` (you may have to enable viewing of hidden files to see it) and look for this line (around line #8):

```
<FilesMatch "(index.php|dl.php|ut.php|lt.php|download.php)$">
```

Change this line to include `archive.php`. The line should now look like this:

```
<FilesMatch "(index.php|dl.php|ut.php|lt.php|download.php|archive.php)$">
```

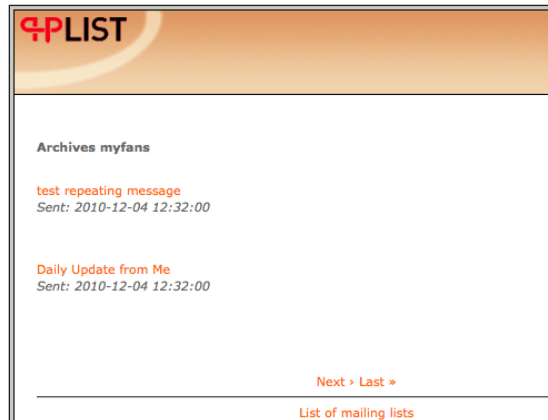
Direct your web browser to <http://your-phplist-installation/archive.php> — you'll see a list of all your lists. Clicking each one will show you a reverse-chronological list of the messages sent to that list, 20 per page:



## Customizing the number of messages displayed per page

By default, 20 messages will be displayed per page. To change the number of messages per page, add the string `"&pagerows=x"` to the URL, where `x` is the number of messages to display.

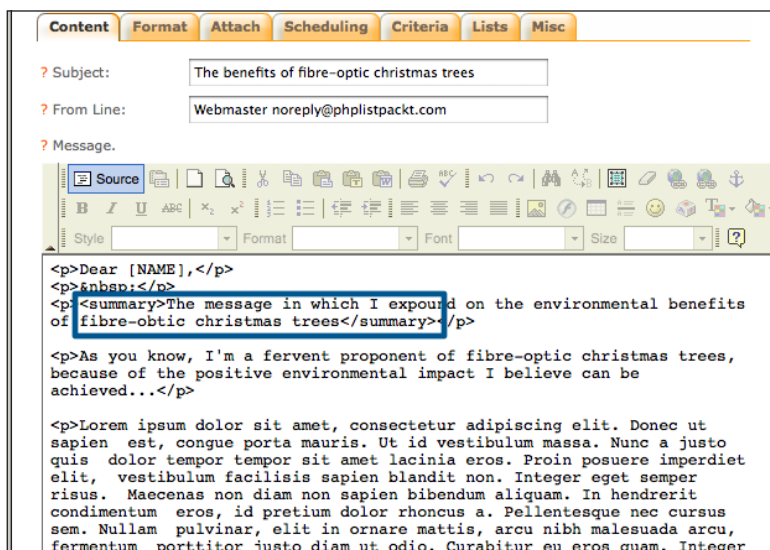
In the following illustration, the string "&pagerows=2" was added to the URL, so only two messages are displayed per page:



## Adding a message summary to the list

If your users are going to be making frequent use of your archives, you may want to include a summary of each message on the **list of messages** page. You can't do this retrospectively to messages that have already been sent, but you can ensure that any subsequent messages include a summary.

In the **send a message** window, when you compose your message, enter the summary between the `<summary>` and `</summary>` tags:







### Adding a summary in HTML source mode

Note that you won't be able to enter the summary tags in the regular WYSIWYG mode. Even if you attempt to do so, the angular brackets will be changed into HTML entities (that is, < becomes &lt; ;), rendering them invalid. You'll need to switch the message editor into Source mode to add these tags.

Once a message with a summary has been processed, the summary will be displayed on the `archive.php` page:



## Hiding the summary in outgoing e-mails

If you don't want the summary text to be displayed in the actual message sent to your readers, encapsulate the summary tags in HTML comment tags. For example, if your hidden summary was to read "Super-subscriber Christmas specials!", you'd enter it as:

```
<!-- <summary>Super-subscriber Christmas specials!</summary> -->
```

## Linking the archives to the main index

Now that you have automatic message archiving, you may want to allow users to reach the archives via a link on the main page.

To accomplish this, open `index.php` and look for the following line (around line #274):

```
printf('<p><a href="./?p=unsubscribe">%s</a></p>', $strUnsubscribeTitle);
```

Directly below this line, add the following:

```
// START HACK - Link to archives
if (isset($strArchiveTitle)) {
    $TitleArchive = $strArchiveTitle;
} else {
    $TitleArchive = "See the archive";
    printf("\r\n" . '<p><a href="archive.php">%s</a></p>',
    $TitleArchive);
}
// END HACK - Link to archives
```

Now the index page will include a link to the archives page:



### Changing the "See the archive" string



If you're using a non-English translation, or if you just want to change the **See the archive** text, set the variable `$strArchive` to the text you want to be displayed in the appropriate language file (under the `texts` subdirectory).

## Sending messages using your e-mail client

One of the limitations of phpList is that you have to use the web interface to compose your messages. Some users may prefer to compose their message using their regular e-mail client or pass on a forwarded message to their list.

There is a phpList hack which implements this functionality. It's described at <http://docs.phplist.com/MailToList>.

## Installing MailToList

To install the MailToList hack, download the latest version (linked to the wiki page, and included with this book) and extract the contents. Copy `m2l_cron.php` to the root of your publicly-accessible phpList directory, and copy the `mailtolist.php` and the `plugins/mailtolist` folder into your `admin/plugins` directory, retaining the folder structure.

Your plugins folder will now look something like this:

```
drwxr-xr-x 2 root root 4096 Apr 24 2010 defaultplugin
-rw-r--r-- 1 root root 265 Apr 27 2006 helloworld.php
drwxr-xr-x 2 root root 4096 Dec 5 18:18 mailtolist
-rw-r--r-- 1 root root 507 Feb 14 2007 mailtolist.php
drwxr-xr-x 2 root root 4096 Apr 24 2010 sidebar
-rw-r--r-- 1 root root 241 Apr 27 2006 sidebar.php
```

Open `admin/plugins/mailtolist/get_email.class.php` and look for the following line (around line #30):

```
var $attach_url = "http://www.yourdomain.com/maillinglist/
attachments/";
```

Change the value of `$attach_url` to the URL which users would use to view your attachments directory, if you have one.

Then look for the following line (around line #34):

```
var $file_path = "C:/wamp/www/phplist test/lists/tmp/";
```

Change the value of `$file_path` to the filesystem path to your attachments directory (the same as `$attachment_repository` in `config/config.php`).

Now open `config/config.php` and look for the following line:

```
define("PLUGIN_ROOTDIR", "/home/me/phplistplugins");
```

Change this to:

```
define("PLUGIN_ROOTDIR", "plugins");
```

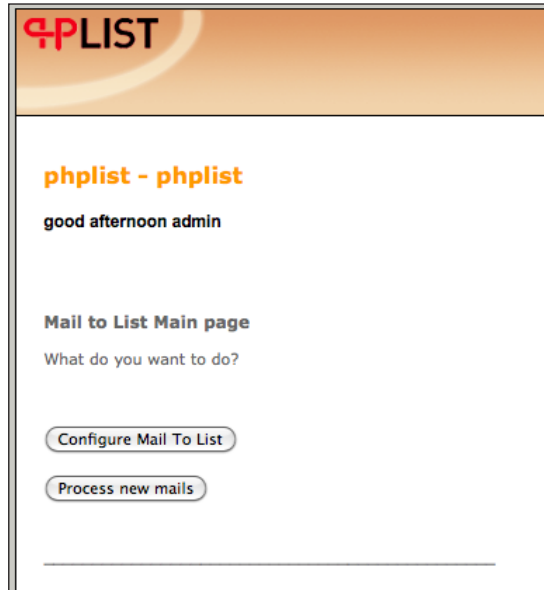


While phpList permits either a relative or an absolute path to be specified as your plugin root dir, the Mail To List cron interface expects this to be a path relative to the /admin/ folder.

Reload the administration web interface. At the bottom of the right-hand sidebar, you'll see new links for the plugins we've enabled. Click on **mail to list**:

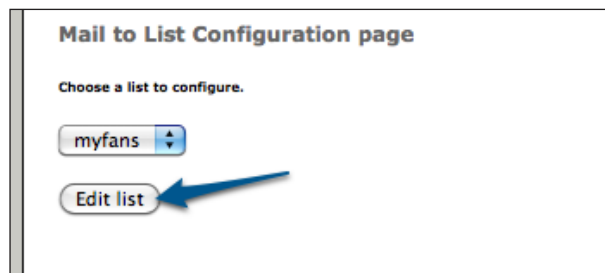
The screenshot displays the phpList administration interface. The main content area shows a message: "Database had to be updated, please reload the page. By the way, everything went fine." Below this message is a button labeled "Click to reload" with a blue arrow pointing to it. The right-hand sidebar contains a menu with various options. At the bottom of the sidebar, there are new links: "hello world", "mail to list", and "mozilla sidebar". A blue arrow points to the "mail to list" link. The footer of the page reads: "© tincan limited | phplist powered by phplist - version 2.10.12".

The first time you click on **mail to list**, the plugin will update your database structure to add its own tables. Click on **Click to reload** to load the "normal" interface you'll see from now on:



## Configuring the Mail To List

Click on **Configure Mail To List** to start the configuration process. For each list to be configured for Mail To List processing, select the list in the drop-down box and click on the **Edit list** button:



For each list, you'll need an independent e-mail address with which to receive incoming messages. Configure your e-mail address, mail box details, and whether or not you want to automatically place incoming messages in the message queue (that is, you may want to manually review them first). Click on **Apply settings** to save:

The screenshot shows the 'phplist - phplist' interface. At the top left, it says 'good afternoon admin'. The main heading is 'Mail to List Configuration page'. Below this, there is a form with several sections:

- Insert the email address for list "**: A text input field containing 'myfans@phplistpackt.com'.
- Select a template to use for sending out your mail**: A dropdown menu with 'No Template' selected.
- POP3 settings:** A table of input fields:

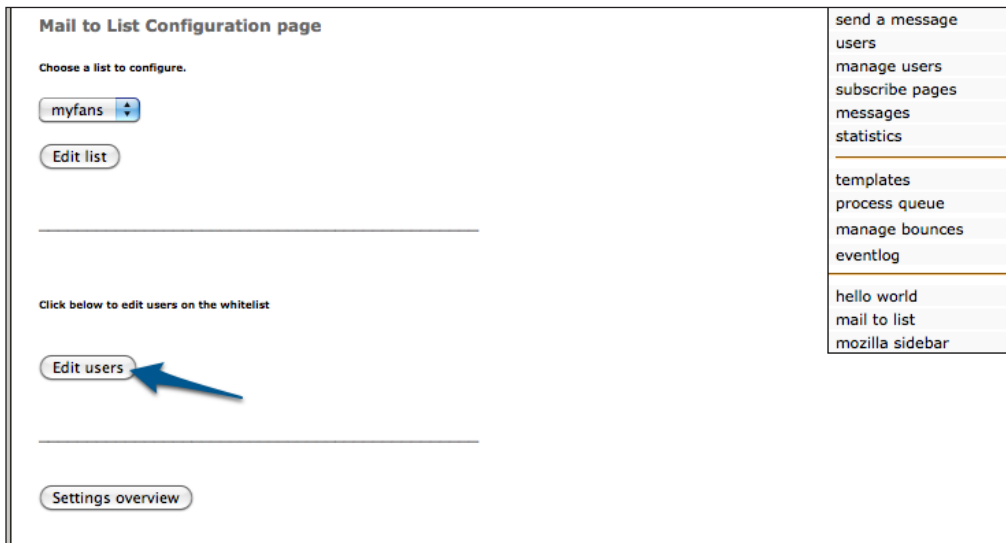
|           |                 |                        |
|-----------|-----------------|------------------------|
| Host:     | localhost       | (mail.yourdomain.com)  |
| Port:     | /pop3:110/notls | /pop3:110/notls)       |
| Login:    | myfans          | (email@yourdomain.com) |
| Password: | myfans          |                        |
- \*Check with your host if you are not sure**: A horizontal line.
- Extra settings:** A dropdown menu for 'Queue messages sent to this address?' with 'Yes' selected.
- Buttons:** 'Apply settings' (highlighted with a blue arrow) and 'Delete settings'.

On the right side, there is a sidebar menu with the following items: main page, configure, help, about, logout, English (dropdown), lists, send a message, users, manage users, subscribe pages, messages, statistics, templates, process queue, manage bounces, eventlog, hello world, mail to list, mozilla sidebar.

Note that you won't be directed back to the main configuration page – you'll need to re-navigate to this page using the sidebar.

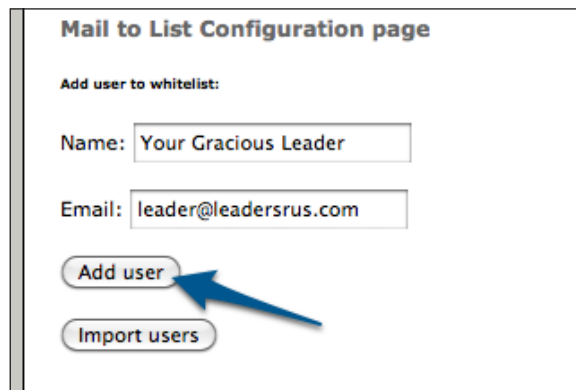
## Whitelisting users

Of course, you don't want just anybody to be able to send e-mails to your list (imagine receiving a spam message and having it re-broadcasted to your trusting subscribers!). To this end, you'll need to "whitelist" trusted e-mail addresses. Click on the **Edit users** button on the main configuration page to whitelist users:



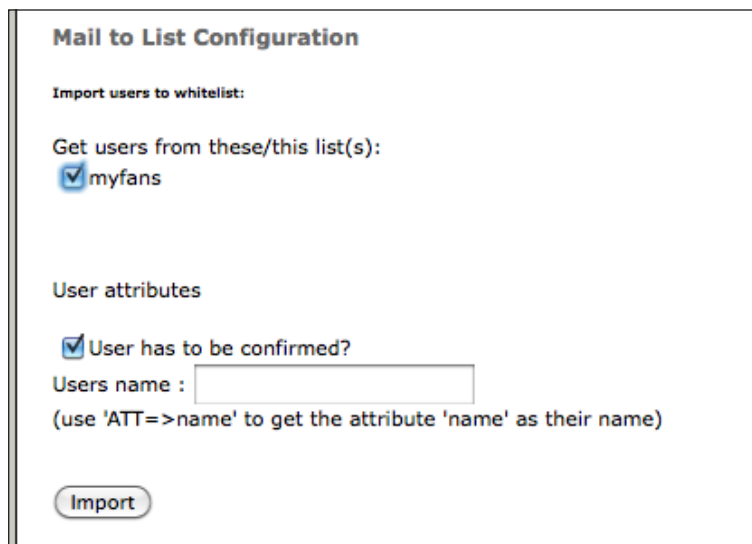
## Manually adding users

To manually add a user, enter their name and e-mail address and click on **Add user**:



## Importing users

If you have a large subscriber base and you want several users to be able to create messages by e-mail, it may be impractical to whitelist them one-by-one. Click on the **Import users** button to begin importing existing list members:



**Mail to List Configuration**

Import users to whitelist:

Get users from these/this list(s):

myfans

User attributes

User has to be confirmed?

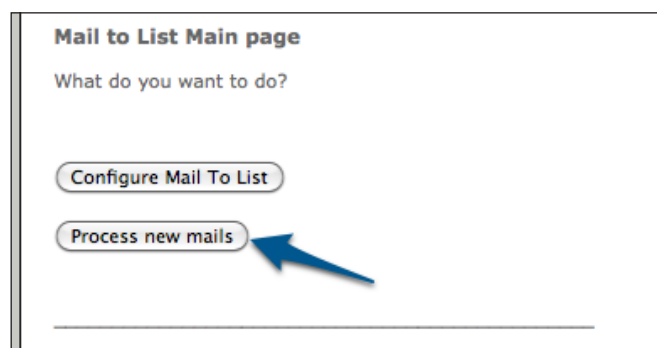
Users name :

(use 'ATT=>name' to get the attribute 'name' as their name)

Select the list from which you want to import users and click on **Import**.

## Processing incoming e-mails

Send an e-mail to the configured address from a whitelisted e-mail account and then click on the **Process new mails** button:

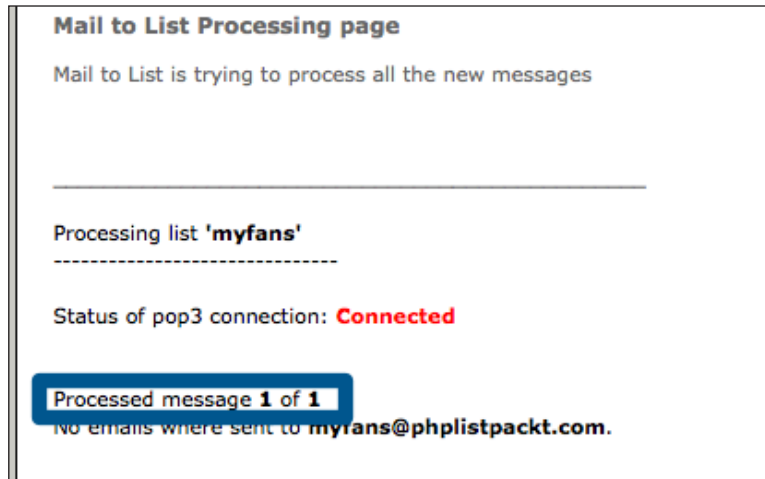


**Mail to List Main page**

What do you want to do?



Note that due to a bug in the code, if there is only a single message to be processed, it will not be reflected as having been successful even though it was (that is, **Processed message 1 of 1** still means that a message was successfully processed):



When you run **process queue** the next time, the message will be delivered as per usual.

## Automating Mail To List using cron

While sending an e-mail to phpList via your e-mail client is very convenient, if you still have to login to manually process the incoming mail box, then you're not saving much time or effort.

Let's automate the processing of this mail box using the `m12_cron.php` file, which you've saved into the root of your phpList folder upon installation earlier.

Using whatever cron-based task scheduling interface you have at your disposal (either via the command line or a web control panel), schedule the running of this file via the PHP CLI binary. That is, if your version is saved in `/home/mywebuser/public_html/lists/m12_cron.php` and your PHP binary is `/usr/bin/php`, then set up a cron task (at whatever frequency you prefer) to execute `/usr/bin/php /home/mywebuser/public_html/lists/m12_cron.php`.

The output of this command will look just like it does in the web interface, except it will be displayed in HTML code, like this:

```
<h1>Mail to List Processing page </h1><p>Mail to List is trying to
process all the new messages</p><br /><br />
-----<br /><br />Processing list <b>'myfans'</b><br />-
-----<br /><br />Status of pop3 connection:
<font color='red'><b>Connected</b></font><br /><br />Processed
message <b>1</b> of <b>1</b><br />No e-mails where sent to <b>myfans@
phplistpackt.com</b>.
```

Combining cron automation with phpList's processqueue automation will, therefore, give you the potential to send messages to your subscribers without having to use the web interface at all.

## Bugs in Mail To List

At the time of writing, the latest version of `mailto` is 2.0.0b. This version suffers from a few bugs which will hopefully be addressed by the developer in future releases. Two such bugs are the following.

### Plain text messages will come out blank

Due to either a flaw in the plugin logic or to changes to recent versions of phpList, if you use your e-mail client to send a message in plain text format, it will be delivered to your subscribers with no content and only the default footer. To work around this, always send your messages in HTML format.

### Plain text or HTML attachments will disappear

Due to the same bit of logic (deconstructing a multipart MIME message), if your attachment is a plain text file or an HTML file, the plugin will assume that file to be the message itself and not the attachment, and confusion will ensue.

## Summary

In this chapter, you have learned to "hack" phpList, both by making minor code edits yourself, using pre-created plugins, or creating standalone files (that is, `archive.php`) to add additional functionality. You "hacked" phpList to add the following features:

- Substituting user attributes in the message subject line
- Sending system messages formatted as HTML instead of plain text
- Creating a messages archive page
- Send messages via your e-mail client

In the next chapter, we'll discuss some common issues encountered using phpList and teach you how to troubleshoot and maintain your installation.

# 11

## Troubleshooting and Maintenance

In the preceding chapters, you've learned about how to use phpList from simple routine list management to more advanced features and settings. In this chapter, you will learn how to maintain, backup, and update phpList, as well as how to troubleshoot common faults and errors.

We will cover the following topics:

- General fault diagnosis
- Common errors and warnings
- Backing up phpList
- Upgrading phpList
- Where to find more help and information

### General fault diagnosis

phpList includes some advanced controls for providing information and debugging when a problem occurs. These can help us determine where the fault lies.

### Test mode (the "dry run")

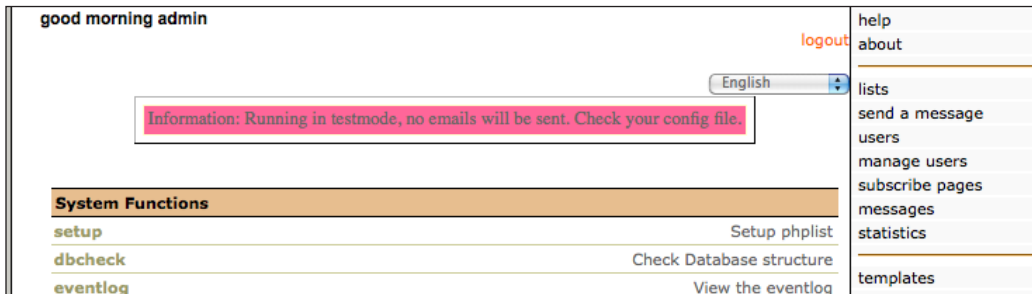
In test mode, phpList will appear to process messages as normal, but will not actually send any messages. This is a useful way to test your configuration and settings without disrupting your subscribers.

Open `config/config.php` and look for the following line (around line #193):

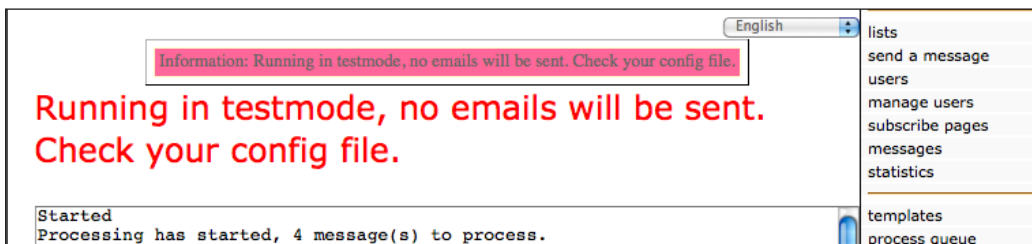
```
define ("TEST",0);
```

Change the value of `TEST` to 1. If `TEST` is set to 1, then no actual e-mail will be sent from phpList.

You will be notified that the phpList's admin interface is in test mode:



Similar messages will be displayed when you process the queue:



## Verbose mode – tell me what you're thinking

In verbose mode, phpList will output far more details than usual when processing messages. For example, it will display every e-mail address used when processing the queue – on a large e-mail list, this can significantly impact the performance of the in-browser "process queue" page!

To enable verbose mode, open `config/config.php`, and look for the following line (around line #198):

```
define ("VERBOSE",0);
```

Change the value of `VERBOSE` to 1 to enable verbose mode.

When sending a message in verbose mode, the process output includes the SQL statements used to determine whether a user should receive a message or not. These can be very helpful in resolving faults around repeat / RSS messages.

|   |  |
|---|--|
| <pre> Processing message 21 Looking for users looking for users who can be excluded from this processqueue run select distinct user.id from   (phplist_listuser as listuser,   phplist_user_user as user,   phplist_listmessage as listmessage) left join phplist_usermessage as usermessage on (usermessage.messageid = 21 and usermessage.userid = listuser.userid) where   listmessage.messageid = 21 and   listmessage.listid = listuser.listid and   user.id = listuser.userid and   usermessage.userid IS NULL   and user.confirmed and !user.blacklisted Found them: 5 to process Sending 21 to herbertneedleman@phplistpackt.com </pre> | <ul style="list-style-type: none"> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> <li>statistics</li> </ul> <hr/> <ul style="list-style-type: none"> <li>templates</li> <li>process queue</li> <li>manage bounces</li> <li>eventlog</li> </ul> <hr/> <ul style="list-style-type: none"> <li>hello world</li> <li>mail to list</li> <li>mozilla sidebar</li> </ul> |
|---|--|

## Common errors and warnings

Here are some common faults encountered with phpList and their resolutions:

### Error: Please make sure that index.php is your default document for a directory

On installing phpList, you see a page entitled **Error: please make sure that index.php is your default document for a directory** as follows:

**Error: please make sure that index.php is your default document for a directory**

If you have just installed PHPlist and get this message, make sure that your Apache configuration has somewhere

```
DirectoryIndex index.php index.html
```

or that at least index.php is mentioned before index.html

For other webserver please consult your manual to find how to make index.php be the default document for a directory.

Alternatively you can delete the file "index.html" in the lists directory of PHPlist

You probably want to be [here](#) or [here](#).

This is a phpList-specific error message intended to appear if:

- You don't have PHP installed
- Your website host prefers to display the contents of `index.html` (a static page) before `index.php` (a dynamic page)

If you are using a web-hosting service, this would be a matter for your provider to resolve.

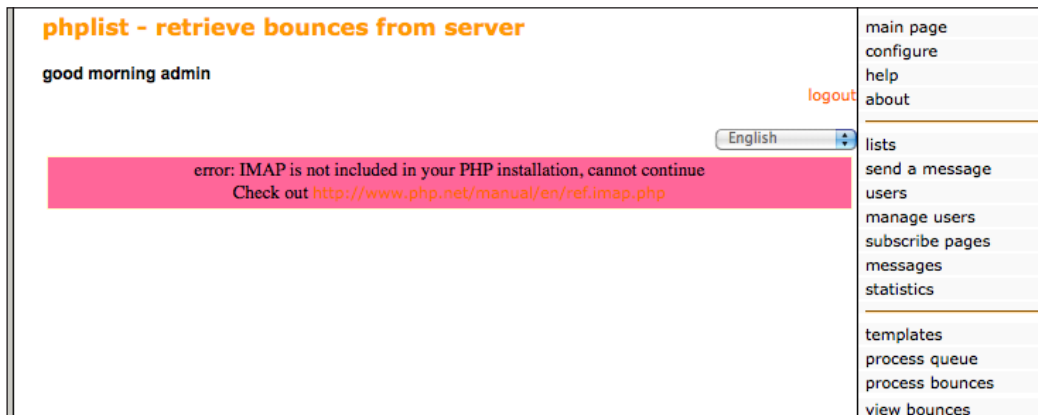
If you manage your own host using Apache, you'll need to edit the configuration for this server and find the `DirectoryIndex` directive. This line tells the web server which file is the "default" page under a given directory. There may be several options in descending order of priority. Ensure that `index.php` is included and that it is included before `index.html`.

The quick-and-dirty way to address this problem may be simply to delete `index.html` so that `index.php` will take priority.

(You can read about the functionality of Apache's `DirectoryIndex` directive at [http://httpd.apache.org/docs/trunk/mod/mod\\_dir.html](http://httpd.apache.org/docs/trunk/mod/mod_dir.html)).

## Error: IMAP is not included in your PHP installation, cannot continue

You may encounter the message **error: IMAP is not included in your PHP installation, cannot continue** when attempting to process bounces, as shown in the following screenshot:



This is because a vital PHP module (php-imap) is missing from your installation. This module is used not only for IMAP connections, but also for POP3 connections and contains the functions that phpList requires to poll the bounce mailbox for new messages.

To resolve this message, either request your web hosting provider to install the php-imap module or install it on your own web server.

## Fatal error: Cannot connect to database, access denied. Please contact the administrator

This error indicates that one of the following four variables in `config/config.php` is incorrect:

- Your database host name (often it's `localhost`)
- Your database name (the actual database on the aforementioned host, where phpList's data is stored)
- Your database username (the user who connects to the aforementioned database)
- Your database user's password (the password used by the aforementioned user to connect)

Confirm each of these either by using a web-based control panel or from a command-line on your host.

If you have command-line access to your web host, you can test these details using the `mysql` command as follows:

```
mysql -h <hostname> -u <user> -p<password> <databasename>
```

A failed attempt will show you a message similar to the following:

```
ERROR 1045 (28000): Access denied for user 'baduser'@'localhost' (using password: YES)
```

A successful attempt will give the following prompt:

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 984
```



Server version: 5.0.77-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>



## HTTP Error 500: Internal server error

The **HTTP Error 500: Internal server error** is the error thrown when *the server encountered an unexpected condition which prevented it from fulfilling the request*. This can be caused by several factors, but in the case of phpList, if you haven't made any changes to the source code, the most likely scenario is that your web server is running PHP as a CGI as opposed to an Apache module.

Some web-hosting providers will prefer to run PHP as a CGI for its security and privilege-separation advantages.

One resolution to this problem in the context of phpList is to edit `.htaccess` in the root of your publicly-accessible phpList files and comment out this line (around line #12):

```
php_flag magic_quotes_gpc on
```

By changing it to:

```
# php_flag magic_quotes_gpc on
```

## HTTP Error 404: File not found

Like the previous Error 500, the 404 Error status is a generic HTTP status meaning that the server couldn't find an appropriate file to match the URL you requested.

phpList needs to know the URL at which it's installed in order to generate appropriate hyperlinks, both on the web interface, and in the e-mails it delivers.

To resolve this fault, confirm that the value of **Website address** on the configure page is correct:

|   |   |
|---|---|
| <p><b>good morning admin</b></p> <p style="text-align: right;"><a href="#">logout</a></p> <p style="text-align: right;">English</p> <p>Here you need to specify the correct values for your website.<br/>You can use certain <b>Placeholders</b> in your values. Placeholders look like <b>[SOMETEXT]</b>, where <b>SOMETEXT</b> can be different. Some useful placeholders are:</p> <ul style="list-style-type: none"> <li>• WEBSITE - the address you type for your website</li> <li>• DOMAIN - the text you type for your domain</li> <li>• SUBSCRIBEURL - the location of the subscribe page</li> <li>• UNSUBSCRIBEURL - the location of the unsubscribe page</li> <li>• BLACKLISTURL - the location of the unsubscribe page for unknown users</li> <li>• PREFERENCESURL - the location of the page where users can update their details</li> <li>• CONFIRMATIONURL - the location of the page where users have to confirm their subscription</li> </ul> <p><b>edit Website address (without http://)</b><br/>phplistpackt.com</p> <hr/> <p><b>edit Domain Name of your server (for email)</b><br/>phplistpackt.com</p> <hr/> <p><b>edit Person in charge of this system (one email address)</b><br/>webmaster@phplistpackt.com</p> | <ul style="list-style-type: none"> <li>configure</li> <li>help</li> <li>about</li> </ul> <hr/> <ul style="list-style-type: none"> <li>lists</li> <li>send a message</li> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> <li>statistics</li> </ul> <hr/> <ul style="list-style-type: none"> <li>templates</li> <li>process queue</li> <li>process bounces</li> <li>view bounces</li> <li>eventlog</li> </ul> <hr/> <ul style="list-style-type: none"> <li>hello world</li> <li>mail to list</li> <li>mozilla sidebar</li> </ul> |
|---|---|

Also confirm the values of `$pageroot` and `$adminpages` in `config/config.php`.



If you've migrated your phpList installation from one URL to another, you may find that you are unable to save any changes made to the configure page, because phpList will redirect you to the previous URL when attempting to save. In this instance, you'll need to use a database editor to manually change the value of the `website` field in the `phplist_config` table in the database.

## Warning: The pageroot in your config does not match the current location

This warning (**The pageroot in your config does not match the current location**) will be displayed on the admin interface if the values of `$pageroot` and `$adminpages` do not correspond to the current URL:

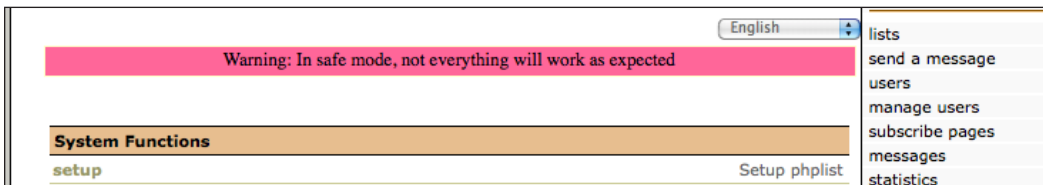
|  |   |
|--|---|
| English  | <ul style="list-style-type: none"> <li>lists</li> <li>send a message</li> <li>users</li> <li>manage users</li> <li>subscribe pages</li> <li>messages</li> <li>statistics</li> </ul> |
| <p><b>Warning: The pageroot in your config does not match the current location</b><br/>Check your config file.</p> |   |
| <p><b>System Functions</b></p> <p>setup <span style="float: right;">Setup phplist</span></p>                       |   |

For example, if you were accessing the admin interface at `http://your-installation/demo/phplist`, but you had configured `$pageroot` as `/lists`, then this error would be presented. It indicates that the URLs generated by phpList in both the configuration section and in any e-mails sent would be invalid.

Confirm that the values of `$pageroot` and `$adminpages` both align with the URL currently used to access the admin interface.

## Warning: In safe mode, not everything will work as expected

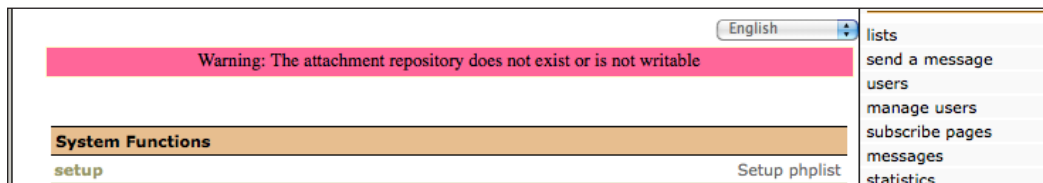
PHP's "safe mode" (*deprecated in upcoming version 6*) was intended to increase the security of the web server by restricting certain PHP functions. This has various restricting effects on phpList, the primary one being that automated bounce list processing is not possible.



If you are seeing this message, either you or your hosting provider should disable safe mode, if possible. If it's not possible to disable safe mode, you will need to process bounces manually. There is also the possibility of problems occurring when sending messages to lists that exceed 500 subscribers.

## Warning: The attachment repository does not exist or is not writable

This message indicates that the filesystem path in which you've configured phpList to save its attachments, either doesn't exist, or is not writable by the web server process:



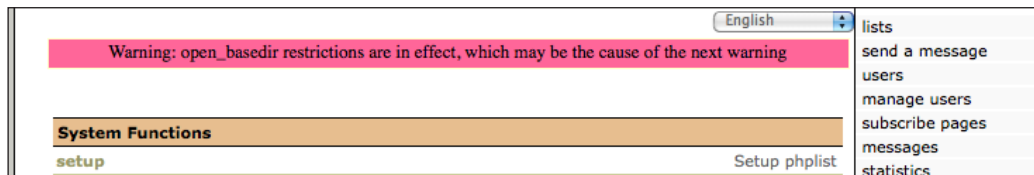
Open `config/config.php` and look for the following line (around line #583):

```
$attachment_repository = '/tmp';
```

Change the value of `$attachment_repository` to match the location (ideally a non-public location) where you want phpList to store uploaded files. Make sure that this directory exists and that the web server process has write-access to it.

## Warning: open\_basedir restrictions are in effect

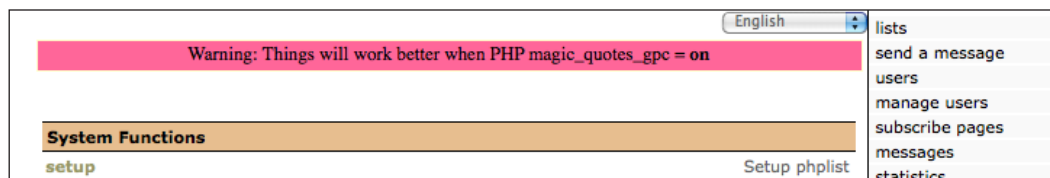
This message suggests that the PHP configuration on your web host has been set up with a specific range of directories that PHP may access. This `open_basedir` setting is often configured on shared hosts for additional security.



This warning is more often found in combination with the previous warning (**The attachment repository does not exist or is not writable**). If you see the two of these messages together, there's a high likelihood that the `open_basedir` security setting is preventing PHP from accessing your chosen `$attachment_repository` directory. Contact your hosting provider to determine which directories are usable under their `open_basedir` configuration", and make sure that your `$attachment_repository` directory is located within one of these usable directories.

## Warning: Things will work better when PHP magic\_quotes\_gpc = on

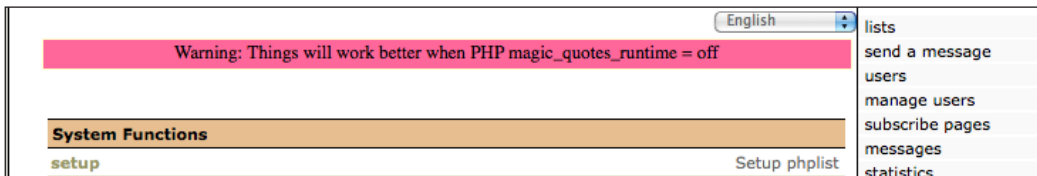
`magic_quotes_gpc` is a PHP setting (*deprecated in version 6*) that automatically escapes incoming data (that is, from HTML forms) to prevent SQL injection attacks. For now, phpList will work better when `magic_quotes_gpc` is enabled and it will present this warning message when it is not:



The `.htaccess` file supplied with phpList will automatically enable `magic_quotes_gpc` if possible. If you are seeing this message, it's likely your PHP configuration has been "locked down" to prevent `magic_quotes_gpc` from being enabled and you will need to contact your web hosting provider.

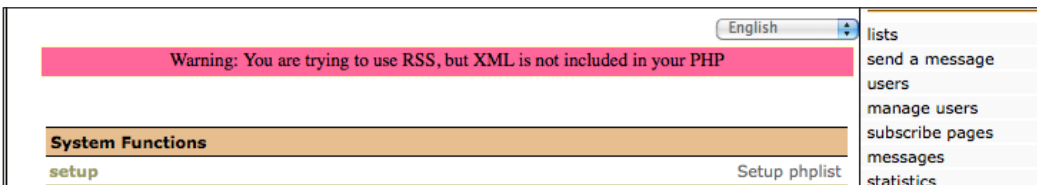
## Warning: Things will work better when PHP `magic_quotes_runtime = off`

In contrast to the `magic_quotes_gpc` feature just discussed, `magic_quotes_runtime` will automatically escape any data returning from an external source at runtime, such as a database query or a filesystem attachment. phpList will also work better when `magic_quotes_runtime` is set to `off`:



## Warning: You are trying to use RSS, but XML is not included in your PHP

You will see this message if you've enabled RSS processing, but your PHP installation doesn't include XML support:



To resolve this, compile PHP with XML support, install the appropriate `php-xml` module, or request your web hosting provider to arrange it.

## Mailer Error: SMTP Error: Could not connect to SMTP host

You may see this error if you've configured phpList to use a custom SMTP host and the PHPMailer component is unable to connect to this host to deliver outgoing e-mails. To check your SMTP host details, edit `config/config.php` on line #670, and ensure that your SMTP server is set correctly. Also ensure that you have connectivity from your phpList installation to this SMTP server (that is, check firewalls and access rules).

```
# To use a SMTP please give your server hostname here, leave it blank
to use the standard
# PHP mail() command.
define("PHPMAILERHOST", 'i.dont.exist.com');
```

## Creating a backup

It's important to create regular backups of your phpList installation, which you can then use to restore it in the event of a data loss or corruption. phpList doesn't have a built-in backup mechanism, but as it's powered by a combination of a database and a collection of PHP and support files, we can back these up manually.

## Backing up the database

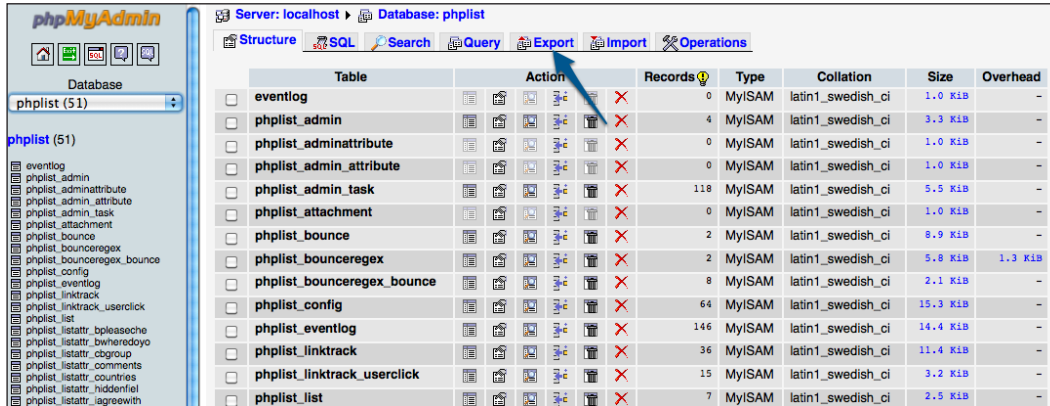
Most web hosting providers will supply an interface to manage your databases, as well as to create backups. This example will illustrate using the popular phpMyAdmin tool to create a database backup.

Log into your phpMyAdmin tool using your phpList database user credentials and select your database from the drop-down list in the left-hand panel.

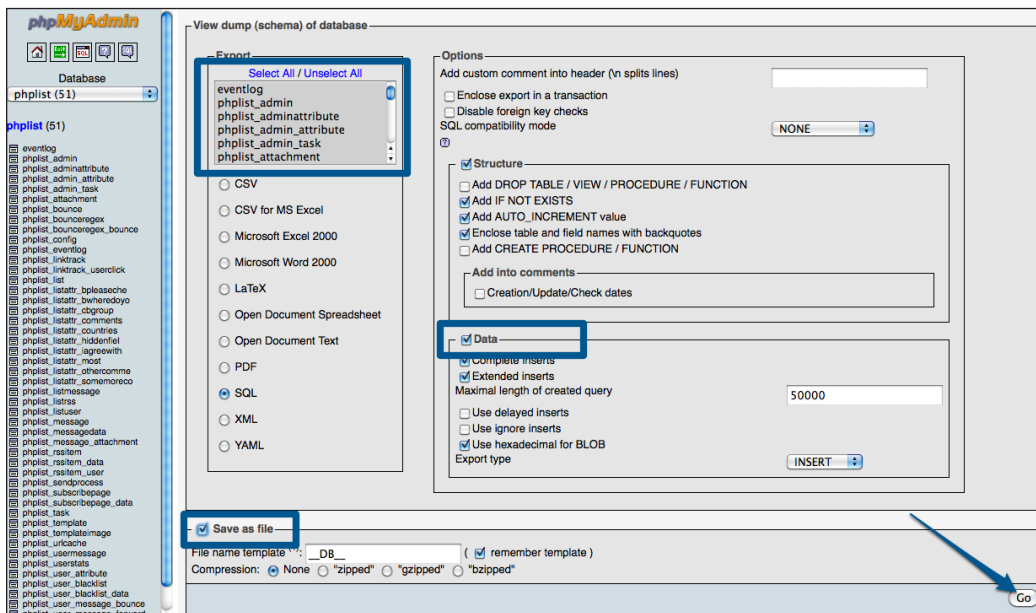


The number of database tables displayed will vary with your installation, as a custom table is created for every attribute you define within your own configuration.

Click on the **Export** tab at the top of the main (right-hand) panel:



On the subsequent export page, ensure that all database tables are selected. Also ensure that the **Data** and **Save as file** checkboxes are ticked and then click on **Go** at the bottom-right:



Your browser will prompt you to download a file. This is a current copy of the database, which can be used to recreate it in the event of a loss.

## Backing up the files

You can simply use your FTP client to download a copy of your phpList folder to another system and save it somewhere for safekeeping.

## Upgrading phpList

Before attempting an upgrade, ensure that you've made a fresh backup of both the database and the phpList files. Specifically, you'll want to ensure you have a copy of `config/config.php` and any other files that you've manually changed.

Download the latest version of phpList from <http://www.phplist.com/download>. Extract it locally and then upload it to your web server, overwriting the existing files.

Once the upload is complete, either copy `config/config.php` over verbatim (for a minor update) or re-customize `config/config.php` using your backed-up copy as a sample.

Load the admin page in your web browser. A message will warn you that your database is out-of-date. Click on the **upgrade** link:

The screenshot shows the 'phplist - main admin page' with the following elements:

- Header: **phplist - main admin page**
- Greeting: **good afternoon admin**
- Language selector: English
- Logout link: [logout](#)
- Error message (pink background):
 

```
Error: Your database is out of date, please make sure to upgrade
Your version : 2.10.8
phplist version : 2.10.12
```
- System Functions table:
 

| System Functions         |                          |
|--------------------------|--------------------------|
| <a href="#">setup</a>    | Setup phplist            |
| <a href="#">upgrade</a>  | Upgrade                  |
| <a href="#">dbcheck</a>  | Check Database structure |
| <a href="#">eventlog</a> | View the eventlog        |
- Navigation menu (right side):
  - [main page](#)
  - [configure](#)
  - [help](#)
  - [about](#)
  - [lists](#)
  - [send a message](#)
  - [users](#)
  - [manage users](#)
  - [subscribe pages](#)
  - [messages](#)
  - [templates](#)
  - [process queue](#)
  - [process bounces](#)
  - [view bounces](#)
  - [eventlog](#)
- Close button: [close](#)



You will be presented with a confirmation message explaining the upgrade process. Click on the word **here** to start the upgrade. This may take some time, depending on the size of your database:



The screenshot shows the phplist admin interface during an upgrade. The page title is "phplist - upgrade phplist". The user is logged in as "good afternoon admin". The interface includes a "logout" link, a language dropdown menu set to "English", and a list of navigation links: main page, configure, help, about, lists, send a message, users, manage users, subscribe pages, messages, templates, and process queue. The main content area displays the following text: "Your database version: 2.10.8", "Your database requires upgrading, please make sure to create a backup of your database first.", and "When you're ready click [here](#). Depending on the size of your database, this may take quite a while. Please make sure not to interrupt the process, once you've started it." A blue arrow points to the word "here" in the text.

Upon completion, you will see an **Information: Success** message:



The screenshot shows the phplist admin interface after the upgrade is complete. The page title is "phplist - upgrade phplist". The user is logged in as "good afternoon admin". The interface includes a "logout" link, a language dropdown menu set to "English", and a list of navigation links: main page, configure, help, about, lists, send a message, users, manage users, subscribe pages, messages, templates, process queue, process bounces, view bounces, and eventlog. The main content area displays the following text: "Your database version: 2.10.8" and "Please wait, upgrading your database, do not interrupt". A pink box with the text "Information: Success" is centered on the page.

To confirm that the upgrade message is no longer displayed and that the database was successfully upgraded, return to the main admin page and then click on the **dbcheck** link to check the database structure (the upgrade link should be gone at this point):

phplist - main admin page

good afternoon admin

logout

English

**System Functions**

setup Setup phplist

dbcheck Check Database structure

eventlog View the eventlog

close

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue

Confirm that the database check is successful (look at the checkmarks):

phplist - phplist

good afternoon admin

logout

English

**Database Check**

| Database Structure  | real name                   | ok | index  | unique  | check                    |
|---------------------|-----------------------------|----|--|---------|--------------------------|
| user                | phplist_user_user           | ✓  | (foreignkey)<br>idx_phplist_user_user_uniqid<br>(uniqid)<br>emailidx (email)<br>enteredindex (entered) | (email) | user open                |
| user_history        | phplist_user_user_history   | ✓  | useridix (userid)<br>dateidx (date)  |         | user_history open        |
| list                | phplist_list                | ✓  | nameidx (name)<br>listorderix (listorder)  |         | list open                |
| listuser            | phplist_listuser            | ✓  | userenteredix<br>(userid,entered)<br>userlistenteredix<br>(userid,listid,entered)                      |         | listuser open            |
| user_blacklist      | phplist_user_blacklist      | ✓  | emailidx (email)   |         | user_blacklist open      |
| user_blacklist_data | phplist_user_blacklist_data | ✓  | emailidx (email)<br>emailnameidx (email,name)  |         | user_blacklist_data open |

- main page
- configure
- help
- about
- lists
- send a message
- users
- manage users
- subscribe pages
- messages
- templates
- process queue
- process bounces
- view bounces
- eventlog

The upgrade is complete.

## Where to find more help and information

phpList is constantly evolving and the features and directions provided in this book will eventually be made obsolete by future versions.

Outside of this book, your primary source of information and assistance will probably be the phpList user forums (<http://forums.phplist.com/>) and the phpList documentation wiki (<http://docs.phplist.com/>).

## Summary

In this chapter, you learned how to troubleshoot common faults and errors with phpList, as well as how to perform routine maintenance on your installation.

You covered the following topics:

- General fault diagnosis
- Common errors and warnings
- Backing up phpList
- Upgrading phpList
- Where to find more help and information

# Index

## Symbols

**\$subject variable** 172  
**.htaccess file** 206  
**.TGZ file** 9  
**[ATTRIBUTE] tags** 171  
**[CONTENT] placeholder** 120

## A

**add a list link** 102  
**add a new one link** 30  
**administrator management**  
  about 105  
  assigning, to lists 109, 110  
  new administrator, testing 109, 110  
  restricted admin, creating 105-108  
  super admin, creating 109  
**admin login screen** 13  
**admin pages**  
  securing, htauth password used 139  
**admin pages security, htauth password used**  
  about 140  
  admin/.htaccess, amending 141, 143  
  htpasswd file, creating 140  
  htpasswd file, creating in Unix shell 141  
  htpasswd file, creating online 140  
  passwords, remembering 143  
**admin password**  
  changing 135, 136  
  strong passwords, choosing 135, 136  
**admin permissions**  
  setting, overview 107  
**admins link** 136

## advanced user/click tracking statistics

  about 130  
  domain statistics, viewing 133  
  overview page 131  
  statistic sidebar link 130  
  view clicks by url link 132, 133  
  view opens by message link 133

## Appearance option 72

### attachments

  adding, to messages 151-153  
  enabling 151  
  files on web server, attaching 155-157  
  multiple files, adding simultaneously 153,  
    154  
  on web server, securing 158

## B

### backup

  creating 207  
  database, backing up 207, 208  
  files, backing up 209

### basic subscribe page

  setting up 28, 29

### bounce

  about 43

### bounce handling

  about 145  
  current rules against bounces, checking 150  
  enabling 145  
  new bounce rule, creating 146, 147  
  new bounce rules, auto-generating 149  
  new rule based creating, existing bounce  
    based 147, 148

- regular expressions 146
- rules 146
- bounce review**
  - e-mail bounce reason, interpreting 49
  - e-mail bounce users, examining 50, 51
  - individual bounce actions 48
  - unconfirmation actions in eventlog, reviewing 51
- bounces**
  - manual processing 46, 47
  - reviewing 47, 48
  - testing 46, 47
- built-in placeholders**
  - [CONTENT] 112
  - [EMAIL] 112
  - [LISTS] 112
  - about 111
- C**
- chmod command 138**
- Clear cached data button 64**
- click tracking**
  - about 125
  - basic statistics per message, examining 129
  - enabled email, sending 128
  - extra detail, enabling 128
  - false spam/phishing detection, avoiding 126-128
- CLI wrapper**
  - customizing 167, 168
- config file 125**
- Configuration functions section 18**
- Configure Attributes item 16**
- Configure General Values 15**
- configure link 16**
- consecutive bounces 45**
- correct filesystem permissions**
  - about 137
  - appropriate permissions, setting 138
  - confirming, in GUI SFTP/FTP client 137
  - shell session, using 138
- Countries in the world attributes 17**
- Create Subscribe Pages 19**
- Criteria tab 115**
- cron**
  - used, for automating Mail To List 194, 195

- custom subscribe page**
  - about 30
  - attributes, creating 30, 31
  - attributes, selecting 34
  - creating 32
  - delivery options, customizing 34
  - HTML footer, customizing 33, 34
  - HTML header, customizing 33, 34
  - list, selecting 35
  - spammer protection 36
  - text, customizing 33
  - title, customizing 33

## D

- dbcheck link 210**
- Default Landing Tab 82**
- default permissions**
  - default permissionssetting 108
- DirectoryIndex directive 200**
- Display subscribe form automatically if only one active list is offered option 29**
- domain-based e-mail throttling**
  - about 151
- downloading**
  - phpList 8
- Drupal-phpList integration. See php List-Drupal integration**
- dry run**
  - general fault diagnosis 197

## E

- e-mail bounce handling**
  - about 43
  - basic settings, configuring 44, 45
  - requirements 43, 44
- e-mail delivery**
  - testing 20
- e-mail delivery, testing**
  - message, sending 22-24
  - message queue(test mode), processing 24
  - test mode, disabling 24
  - user, creating 20-22
- e-mail personalization**
  - about 111
- edit link 28**

## errors

- about 199, 200
- fatal error 201
- HTTP Error 404 (File not found) 202, 203
- HTTP Error 500 (Internal server error) 202
- mailer error 207

## F

### Facebook

- phpList, integrating with 74

**Facebook-phpList integration.** *See* **phpList-Facebook integration**

**Format tab** 119

## G

### general fault diagnosis

- about 197
- test mode 197, 198
- verbose mode 198, 199

## H

**history link** 88, 125

**htaccess restrictions** 139

### HTML code, subscribe form

- customizing 39
- JavaScript, removing 39
- optional pop-up configuration 39
- subscribe form button, removing 40
- tag, changing 39

**htpasswd command** 141

## I

### individual user management

- bulk user management 90
- complex list, importing 93-97
- list memberships, updating 87
- simple list, importing 91-93
- user details, updating 87, 89
- user history, examining 88, 89
- users, creating 86
- users, exporting 100-102
- users, importing 90
- users, reconciling 97-99

- user unsubscribing 88

**Info tab** 83

### initial configuration

- about 10
- database settings 10
- path settings 10

### initial login

- Admin Password, modifying 14
- Attributes, configuring 17
- General Values, configuring 16
- List, creating 19
- Subscribe Pages, creating 20

### installation, phpList Integration Module

- about 54
- attribute mapping 57
- debugging, turning on 61
- Drupal's cache, clearing 63
- Drupal permissions, configuring 62
- external configuration 55-57
- list access, configuring 61
- miscellaneous 60
- My Account-My Newsletter options 58
- My newsletters access, confirming 64
- PHPList subscribe block, adding 65, 67
- PHPList subscribe block, troubleshooting 68
- Registration page options 59
- users, synchronizing to phpList 60, 61

### installation, phpList Integration plugin

- admin login 70
- admin password 70
- form setting 70
- general setting 70
- HTML comments 72
- page, adding 71, 72
- widget, adding 72, 74

### installing

- phpList databases 7

## L

**list administrators page** 105

### lists

- creating 102, 103
- managing 102
- member operations, performing 103-105

lists link 19  
localhost 10  
ls -l command 138

## M

**Magic\_quotes\_gpc** 205  
**magic\_quotes\_gpc** feature 206  
**MailToList**  
  automating, cron used 194, 195  
  bugs 195, 196  
  configuring 190, 192  
  installing 188  
**main page link** 15-19  
**messages**  
  attachments, adding 151-153  
  auto-generating, from RSS feeds 160, 161  
  available criterion amount, increasing 116  
  enhancing, built-in placeholders used 111,  
  112  
  personalizing, member attributes used 113,  
  114  
  placeholders 113  
  repeating, automatically 158  
  repeating message, forcing to repeat 159,  
  160  
  repeating message, sending 159  
  sending, to attribute based subsets 115, 116  
  summary, adding to list 185, 186  
**messages archive page**  
  creating 183, 184  
  linking, to main index 186  
  messages displayed, customizing 184  
  message summary, adding to list 185, 186  
  summary in outgoing e-mails, hiding 186  
**message sending, e-mail client used**  
  about 188  
  incoming e-mails, processing 193, 194  
  Mail To List, bugs 195  
  Mail To List, configuring 190, 191  
  MailToList, installing 188  
  Mail To List automating, cron used 194, 195  
  users, adding manually 192  
  users, importing 193  
  users, whitelisting 192  
**messages link** 123  
**message templates**

  about 117  
  creating 117, 118  
**mod\_auth** level 143  
**My newsletters tab** 64  
**mysql** command 201

## N

**newsletter effectiveness**  
  about 121  
  privacy note 121  
  transparency note 121

## O

**open\_basedir** setting 205  
**overview link** 133

## P

**php-imap** module 8  
**phpList**  
  about 7  
  basic concepts 27  
  documentation wiki 211  
  downloading 8  
  e-mail bounce handling 43  
  errors 199  
  files, installing 7  
  references 211  
  securing 135  
  unpacking 9, 10  
  upgrading 209-211  
  uploading 9, 10  
  user forums 211  
  user tracking 122  
  warnings 199  
**phpList, securing**  
  admin pages, securing 139  
  admin password, changing 135  
  correct filesystem permissions, confirming  
  137  
  htaccess restrictions, confirming 139  
**phpList-Drupal integration**  
  about 53  
  phpList Integration Module configuration  
  54, 55

- phpList Integration Module installation 54, 55
  - prerequisites 54
  - phpList-Drupal integration plugin**
    - phpList Integration Module configuration 69, 70
    - phpList Integration Module installation 69, 70
  - phpList-Facebook integration**
    - App, creating 77, 79
    - Facebook page, configuring 80, 82
    - list, preparing 75, 76, 77
    - phpList, as default tab 82, 83
    - prerequisites 74
  - phpList-WordPress integration**
    - about 68
    - phpList Integration plugin configuration 68
    - phpList Integration plugin installation 68
    - prerequisites 68
  - phplist - templates in the system page 118**
  - phpList database**
    - installing 7
  - phpList files**
    - installing 7
  - phpList files installation**
    - creating 10, 11
    - initial configuration, performing 10
    - initial login 13
    - requirements 8
    - steps 7
    - web-based configuration, performing 11
  - phpList Integration Module installation.**
    - See installation, phpList Integration Module*
  - phpList Integration plugin installation.**
    - See installation, phpList Integration plugin*
  - PHPList Integration widget 74**
  - PHPListMailer class 176**
  - phplist setup link 12**
  - pop access 44**
  - popuser 45**
  - predefined defaults link 17**
  - processing automation**
    - CLI wrapper, customizing 167, 168
    - requirements 167
    - setting 166
  - public\_html folder 9**
- ## R
- recaptchalib.php 36**
  - reconcile users link 98**
  - regular expressions 146**
  - requirements, phpList files installation**
    - list 8
  - RSS**
    - feed, associating with list 162
    - feed, messages auto-generating from 160
    - messages, user requirements for receiving 163
    - messages schedule, setting 165, 166
    - new RSS items, getting 163
    - support, enabling 161
    - used, for sending messages, sending 165
  - ruleflow. *See* Drools Flow**
- ## S
- Save Changes button 119**
  - send a message link 22**
  - setup link 15, 16**
  - Set Up New App button 77**
  - show link 48**
  - slashes,**
    - stripping, from database stored messages 176
    - stripping, from confirmation message 177
    - stripping, from member data change notifications 177
    - stripping, from personal location message 178
    - stripping, from unsubscribe message 177, 178
    - stripping, from subscribe message, for normal subscriptions 177
    - stripping, from subscribe message, for user imports 178, 179
    - stripping, from subscribe message, when resending confirmations 179
  - slashes, stripping from messages**
    - web interface changes 179



## **spammer protection, subscribe page**

- about 36
- admin/subscribelib2.php, modifying 37
- index.php, modifying 36
- PHP library, downloading 36
- reCAPTCHA keys, signing up for 36

## **standard system messages, web interface changes 179, 180**

### **subscribe form**

- embedding, within other application 38, 39
- HTML code, customizing 39
- HTML code, generating 38

### **subscribe pages list 29**

### **system messages**

- sending, as HTML 174

### **system messages, sending as HTML**

- code changes 174
- CSS styling 183
- HTML, stripping out 175
- HTML messages, composing 182
- message HTML version, sending 175, 176
- message plain-text part, creating 175
- message text version, sending 175, 176
- placeholders 182, 183
- slashes, stripping from confirmation message 177
- slashes, stripping from member data change notifications 177
- slashes, stripping from messages 176
- slashes, stripping from personal location message 178
- slashes, stripping from unsubscribe message 177
- slashes stripping from subscribe message, for normal subscriptions 177
- slashes stripping from subscribe message, for user imports 178, 179
- slashes stripping from subscribe message, when resending confirmations 179
- tips 181, 182
- web interface changes 179

## **T**

### **Turn on debugging checkbox 61**

## **U**

### **unsubscribe page link 88**

### **update page link 87**

### **user attributes**

- forward message 174
- no attribute substitution, for third party 173
- substituting, in subject line 171, 172

### **users**

- all users, exporting 100
- bulk management 90
- complex list, importing 93-97
- details, updating 89, 90
- exporting 100
- history, examining 88, 89
- importing 90
- individual management 86
- reconciling 97-99
- simple list, importing 91, 92
- specific list, exporting 101, 102
- unsubscribing 88

### **users link 20**

### **user tracking**

- about 122
- enabling, on message 122
- statistics, examining per message 123, 124
- statistics, examining per user 124
- turning on, by default 123

## **V**

### **view bounces link 47**

### **view clicks by message link 132**

### **viewed counter 131**

### **View page link 72**

### **view users link 129**

## **W**

**warning** 203-206

**web-based configuration**

initial database 11, 12

performing 11

**web interface changes**

confirmation messages, on pre-existing  
subscribe page 181

standard system messages 179, 180

subscription messages, on pre-existing  
subscribe page 181

**web server**

attachments, securing 158

files stored, attaching 155, 156

**Website address attribute** 16

**WordPress-phpList integration.** *See*  
phpList-WordPress integration

## **Z**

**ZIP file** 8



## Thank you for buying **phpList 2 E-mail Campaign Manager**

### **About Packt Publishing**

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: [www.packtpub.com](http://www.packtpub.com).

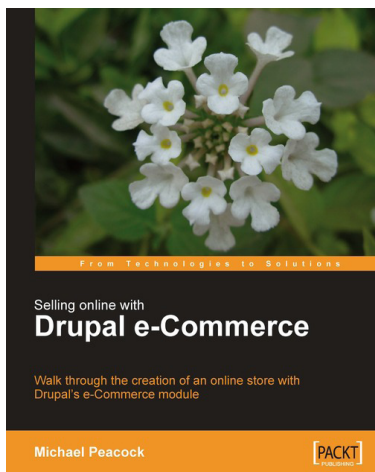
### **About Packt Open Source**

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

### **Writing for Packt**

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to [author@packtpub.com](mailto:author@packtpub.com). If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.



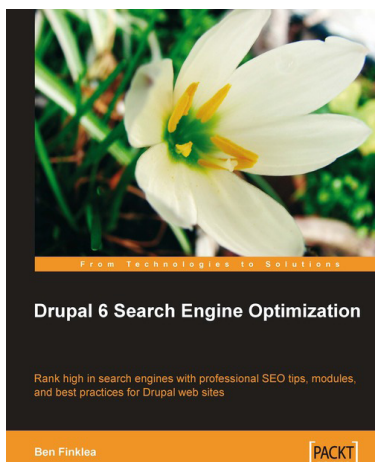
## Selling Online with Drupal e-Commerce

ISBN: 978-1-847194-06-0

Paperback: 264 pages

Walk through the creation of an online store with Drupal's e-Commerce module

1. Set up a basic Drupal system and plan your shop
2. Set up your shop, and take payments
3. Optimize your site for selling and better reporting
4. Manage and market your site



## Drupal 6 Search Engine Optimization

ISBN: 978-1-847198-22-8

Paperback: 280 pages

Rank high in search engines with professional SEO tips, modules, and best practices for Drupal web sites

1. Concise, actionable steps for increasing traffic to your Drupal site
2. Learn which modules to install and how to configure them for maximum SEO results
3. Create search engine friendly and optimized title tags, paths, sitemaps, headings, navigation, and more
4. A practical, step-by-step guide that takes the mystery out of Drupal SEO

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles