



phpAuthent

version 0.2.1

Installation and configuration guide

Written by Arnaud Vincent (vincentarn@gmail.com)

Last updated on August 28, 2005

Table of Contents

1 Installation.....	3
1.1 Downloading phpAuthent.....	3
1.2 Package extraction.....	3
1.3 Installation.....	3
1.3.1 Note about passwords encryption.....	3
1.3.2 Database setup.....	4
1.3.3 Edit the configuration file.....	4
1.3.4 Upload.....	4
1.3.5 First steps.....	4
2 Upgrading from a previous version.....	6
2.1 Backup.....	6
2.2 Upgrade SQL scripts.....	6
2.3 Update configuration.....	6
3 Configuration.....	7
3.1 Database settings.....	7
3.2 Redirections / Files location.....	7
3.3 Admin group security scope.....	8
3.4 Session lifetime.....	8
3.5 Language.....	8
3.6 Passwords generation.....	8
3.7 Demonstration mode.....	8
3.8 Passwords encryption.....	9
3.9 Other settings.....	9
4 Securing web pages.....	10
4.1 General include.....	10
4.2 Login and logout actions.....	10
4.2.1 Login page.....	10
4.2.2 Logout action.....	12
4.3 phpAuthent API functions.....	12
4.3.1 Determining if a user is connected to the application.....	12
4.3.2 Getting the login name of a connected user.....	12
4.3.3 Page level security functions.....	12
4.3.4 In-page security function.....	13
5 Feedback, bugs and comments.....	15
6 Licence.....	16
7 News and updates.....	22

1 Installation

*This chapter describes how to download, install and briefly configure to get *phpAuthent* working on your system.*

*Installation process instructions are much detailed in this documentation, but getting *phpAuthent* up and running should be an matter of minutes.*

1.1 Downloading phpAuthent

phpAuthent is released regularly through SourceForge File Release System at http://sourceforge.net/project/showfiles.php?group_id=134101. You can find this link more conveniently by browsing the 'Downloads' section at <http://phpauth.sf.net>.

As of release 0.2.0, files are released both in zip format (Windows based platforms) and tar.gz (Unix like platforms).

The package contains the following:

- Core engine of phpAuthent
- Database schema for creating or updating MySQL tables and the default administrator user
- phpAuthentAdmin administration web-based interface.

1.2 Package extraction

Extract the content of the downloaded archive somewhere on your local file system. The archive extracts a directory called *phpauthent*.

1.3 Installation

For a fresh new installation of *phpAuthent*, please complete the following first steps to configure your *phpAuthent* application **before uploading to your web server**.

Note

The main configuration file of *phpauthent* is called *phpauthent_config.php* and is located directly under the *phpauthent* directory you've extracted from the archive.

You'll have in these steps to base configure phpAuthent (security, database settings), upload the files to your webserver, and change the administrator password through the web administration console.

1.3.1 Note about passwords encryption

For security reasons, users passwords are encrypted in the database. The encryption is based on a **key** that is stored into your *phpauthent_config.php* file (variable *\$phpauthent_enckey*). By default, this key has the value
Ex6wCoVjh80Iu7ZAraanEEUyJmPHjCI

The '*administrator*' user stored in the database creation file is default assigned a password that is created **with this key**.

Important

You have the possibility to change this encryption key. For this, please follow this 2 steps wizard to choose a new key : http://phpauth.sourceforge.net/wizard_step1.php.

Please note that you won't be able to change the key after this step ! (You will be redirected to here after the operation).

1.3.2 Database setup

phpAuthent requires a valid MySQL database to store users, groups and relationships between both. Tables used by phpauthentic are prefixed with *phpauthentic_*, allowing you to use both an existing database to store *phpAuthentic* data, or dedicate a new database for it.

A script residing in the *sql directory* allows you to create automatically the necessary tables and primary data required by *phpAuthentic* to run. Prior executing this script, you need to run a MySQL client (phpMyAdmin, MySQL Administrator, or others) to:

- **Create a new database** and select it

or

- **Select an existing database**

Then, on the selected database, **run the phpauthentic.sql script**. The execution of this script will create 3 tables:

- *phpauthentic_users*
- *phpauthentic_groups*
- *phpauthentic_relation*

and create an 'administrator' user (group 'admin', password 'phpauthentic').

and execute using the *phpauthentic.sql* file that is located the *sql directory* of your extracted package (You may have modified it earlier if you changed the encryption key).

1.3.3 Edit the configuration file

Edit the *phpauthentic_config.php* file to configure access to your database.

```
// Database (MySQL) settings
$db_hostname = 'localhost'; // Hostname of the MySQL server if not local
$db_username = 'your_username'; // Database access username
$db_password = 'your_password'; // Database access password
$db_database = 'database_name'; // Used database name
```

1.3.4 Upload

First steps configuration are over. You can upload phpauthentic to your web directory.

1.3.5 First steps

1. Point your browser to <http://localhost/phpauthentic/> (assuming that localhost is the location of your web server and that you uploaded phpauthentic to the root directory).
2. You should be redirected automatically to the sample login page with an error message "Access Denied". Login with the user 'administrator' using as a password 'phpauthentic' You will be redirected to </phpauthentic/index.php> which presents a link to the web administration console and a few explanations. Note : the *index.php* page is only here to protect the content of the phpauthentic directory. You are free to modify it, but keep in mind that this directory, as it contains your configuration, must be secured !

3. Go to the administration panel. First, change the administrator password by editing the user and then changing the password. You can now create users, groups, and attribute users in groups to prepare your site authentication schema. Read the following chapters to know what to do in your existing web pages.

2 Upgrading from a previous version

This chapter describes how to upgrade phpAuthent from a previous release. hpAuthent keeps has an upgrade that allows to migrate from one version to another. The upgrade process is realised step by step (version by version), and you need to apply upgrade patches to your existing installation for each release existing between the one you currently have installed and the one you wish to install.

Please follow the next steps to upgrade.

2.1 Backup

- **Backup** your existing **phpAuthent database / tables**
- **Backup** your current **phpauthent_config.php** file
- **Extract** the content of the **newest phpauthent package over the old one.**

2.2 Upgrade SQL scripts

Apply the necessary upgrade sql script(s). Those scripts are located in the sql subdirectory of the phpauthent package.

For example, if you're upgrading from 0.1.3 to 0.2.0, you need to apply the update-0.2.0.sql script.

2.3 Update configuration

Re-apply your the settings from your backup phpauthent_config.php file to the newly written.

Please take a look at the changelog or the eventual release notes for new configuration settings to setup before testing your application.

3 Configuration

The whole configuration of phpAuthent is based on the edition of the file `phpauthent_config.php` file. This file is located in the root directory of the core package.

3.1 Database settings

Reach the following section in the configuration file:

```
// Main configuration
// Database (MySQL) settings
$db_hostname = 'localhost';
$db_username = 'your_username';
$db_password = 'your_password';
$db_database = 'database_name';
```

Replace the fields to reflect your configuration. Things you must specify are:

1. The database hostname on which runs MySQL
2. The database username used to connect to MySQL
3. The database password for this user
4. The database name in which are stored phpAuthent tables.

3.2 Redirections / Files location

```
// Files location
```

```
// The paths specified below has to be defined as full paths (beginning with
a / designing the root
// of your webserver space.
// Path to the login page.
$phpauth_login = '/phpauthent/phpauthent_login.php';
// Target of a successful login
$phpauth_successfull_login_target = "/phpauthent/index.php";
// Target of a successful logout
$phpauth_successfull_logout_target = "/phpauthent/index.php";
```

Those 3 settings indicate the core engine what to do after logins and logouts. Those path need to be full path (not relative) from the document root of your web space (the path must begin with a '/').

- The first setting (`$phpauth_login`) indicates the engine where is located the login page (please read chapter 2.2.1 in this guide for more information). It is used when a login is required to access a page.
- The second setting (`$phpauth_successfull_login_target`) indicates the engine on which page to redirect after a successful login. If you want to go to you web index instead for example, you would set `$phpauth_successfull_login_target="/index.php"`. Please notice that when the login page is automatically shown after a bad access attempt, the successfull login automatically redirects to the page that requested this login. This parameter has only effect when someone has requested for the login page directly.

- The third setting (`$phpaut_successfull_logout_target`) indicates the engine on which page to redirect after a logout.

3.3 Admin group security scope

You can here define whether or not admin users can always be enabled. This prevents specifying the 'admin' group in each access definition (see below how to define accesses for pages / page elements).

```
// Admin security scope
// When setting admin_always_enabled to true, a user member of the admin
group can pass through any
/// security setting (similar to setting the 'admin' group in every
groupsList array)
$admin_always_enabled = true;
```

3.4 Session lifetime

Not used - Prepared for future releases.

```
// Session lifetime
// Specify here the number of seconds after which the cookie session handler
expires to force a new login
$session_lifetime      = 300; // IN DEVELOPMENT - NOT USED
```

3.5 Language

```
// Language
// Specify here the language file (in the phpauthadmin/locale directory)
which suits to your language.
$phpauth_language = 'english.php';
```

This concerns the languages available for the phpAuthentAdmin application (administration interface). The languages settings concern :

- Eventual error or information messages
- Translation of the administration interface

If you can / want to contribute by translating to your native language, please email me at vincentarn@users.sourceforge.net or submit the new language file as a patch through the project page.

3.6 Passwords generation

Not used - Prepared for future releases.

```
// User creation, password wizard (not implemented)
$enable_password_wizard = false;
$passwords_chars_length = 8;
```

3.7 Demonstration mode

```
// Demo mode
// When set to true, no edition is possible (rename, delete, assign or
create)
$demo_mode_enabled = false;
```


The engine can be ran in demo mode so that overview is more significant (for example in this presentation web site...). The demo mode has for effect that nothing could be written into the database.

3.8 Passwords encryption

```
// Encryption secret key
$phpauthent_enckey = "Ex6wCoVjh80Iu7ZAraanEEUyJmPHjCIt";
```

This is the key to use for encrypting passwords in the database (introduced in release 0.1.2). Please notice that if you change it after having created the database, you will have to replace users passwords (including the default 'administrator' user's password) in the phpauthent_users table directly. [A wizard to choose and change the key is available online here \(http://phpauth.sourceforge.net/wizard_step1.php\)](http://phpauth.sourceforge.net/wizard_step1.php).

3.9 Other settings

To be documented

4 Securing web pages

4.1 General include

To enable phpAuth functionalities in your web pages, you only have to include the following code at the very top of your source code.

```
<?php
    require_once("phpauth_core.php");
?>
```

Note

The above code assumes that the phpauth installation files are accessible in the same directory than the edited page. Update the path to `phpaut_core.php` to reflect the changes.

If you can, use the `$_SERVER['DOCUMENT_ROOT']` server variable to locate `phpauth_core.php` file. For example, if phpAuthent is installed in the `phpauth` subdirectory of your dedicated web space, replace the above code by :

```
<?php
    require_once($_SERVER['DOCUMENT_ROOT']."phpauth_core.php");
?>
```

4.2 Login and logout actions

4.2.1 Login page

The login page is used to present users a form with username and password, so that they could authenticate for the rest of the site. phpAuthent contains a sample login page (`phpauth_login.php`) in the sample application. You are free either to customize it to suits your site layout, or implement a complete new page. To build a new login page, only follow the next few steps:

1. Include the `phpauth_core.php` and the `phpauth_config.php` files at the very top of the php page as illustrated below.

```
<?php
    require_once("phpauth_core.php");
    require_once("phpauth_config.php");
<?>
```

2. Build an HTML form, and fill the action attribute as follow :

```
<form name="loginform" method="post" action="<?=$phpauth_loginform_action?>">
    ...
</form>
```

Please note the the form method has to be set to "post".

3. Place an HTML input tag (type "text") to handle the username entry. Fill its name attribute as follow :

```
<input name="<?=$phpauth_loginform_username?" type="text" id="username">
```

4. Place an HTML input tag (type "password") to handle the password entry. Fill its name attribute as follow :

```
<input name="<?=$phpauth_loginform_password?" type="password" id="password">
```

5. Place an HTML input tag (type "submit") to place a submit button. That's it, your form should work this way.

6. Optionally, you can set up the interpretation of the error message within the login page. A wrong login returns an error code 011 through the GET method (err=011 appears in the URL). For this, just paste the following code at the place you want the message to appear :

```
<?php
    if (isset($_GET['err']) && ($_GET['err'] == '011')) {
        print "<p><strong>Invalid username or password</strong></p>";
    }
?>
```

Note : When the login page is requested automatically due to a page protect, a successful login automatically redirects to the page that was initially requested.

7. You can also display a message in the login page when this one is shown automatically after a bad access attempt (i.e. an access to a page for which protection requires to be logged on or a specific user/group). When the login is called in such a way, an error code 012 is set in the GET (err=012 appears in the URL). To display a message, you could set up this kind of code :

```
<?php
    if (isset($_GET['err']) && ($_GET['err'] == '012')) {
        print "<p><strong>Access denied. Authentication required</strong>
</p>";
    }
?>
```

When a user is logged and is rejected from a page due to non sufficient access rights, the login page is also shown with an error code 019.

```
<?php
    if (isset($_GET['err']) && ($_GET['err'] == '019')) {
        print "<p><strong>Access denied. You do not have necessary
authorizations</strong></p>";
    }
?>
```

Login action returned error codes

err=011	Invalid username or password
err=012	Access denied. Authentication required
err=019	Access denied. Insufficient access rights

4.2.2 Logout action

To implement the logout action, simply create a link (HTML `<a>` tag) that points the following URL : `phpauth_core.php?action=logout`. When clicking on this link, user will be disconnected from his session and redirected to the page specified by the configuration parameter `$phpauth_successfull_logout_target` (in the `phpauth_config.php` file). Example:

```
<a href="phpauth_core.php?action=logout">Click here</a> to logout
```

4.3 phpAuthent API functions

4.3.1 Determining if a user is connected to the application

The function `isUserLogged()` gives you the information if a user is connected to the application. The usage of such a function is, for example, to present a "Login / Connect" or "Logout / Disconnect" button depending on the status given by `isUserLogged()`. Example:

```
<?php
    if (isUserLogged()) {
?>
<p class="style1">A user is currently connected to the application as
<strong><?=getUsername()?></strong> (<a href="?action=logout">Click here</a>
to logout)</p>
<?php
    } else {
?>
    <p class="style1">There is curently <strong>no user </strong>connected to
the application. </p>
<?php
    }
?>
```

4.3.2 Getting the login name of a connected user

You can retrieve the name of the user that is connected to the application. For this, use the `getUsername()` function. If no user is logged, the `NULL` value is returned. An example is shown in the code sample above.

4.3.3 Page level security functions

There are two ways to secure a page using Page Level Security.

- **Simple page level security** blocks a page to any user that is not logged into the application. Then, the only security need to be able to view this page content's is to be succesfully logged in the application. To enable Simple page level security in your pages, simply add the following code at the top of your source code:

```
<?php
    require_once("phpauth_core.php");
```

```
simplePageProtect();  
?>
```

- **User/Group page level security** blocks a page to any user that do not belong to the groups or users list that you specify in the page code.

To enable the user/group page level security in your pages, add the following code at the top of your source code:

```
<?php  
    require_once("phpauthent_core.php");  
    $usersArray = array("test_sample_2");  
    $groupsArray = array("test");  
    pageProtect($usersArray,$groupsArray);  
?>
```

The function `pageProtect($usersArray,$groupsArray)` is the explicit call to the page protection. Before calling this function, you need to define two arrays, one for the users and one for the groups. For example, if you want the users `user1`, `user2`, and `user4` to have access to this page, you need to specify the following array and pass it as a first argument of the `pageProtect()` function.

```
$usersArray = array("user1","user2","user4");
```

If you want the members of groups `family` and `friends` to have access to this page, you need to specify the following array and pass it as the second argument of the `pageProtect` function.

```
$groupsArray = array("family","friends");
```

Note : You're free to name the users and groups arrays as you want.

4.3.4 In-page security function

With **in-page security**, you can present on a public page (all users can access the page) user -or group- specific content depending on the user logged into the website. The user then authenticates to be able to see additional information.

Let's imagine that you have a 'photo' page that you want to be the access point to all your pictures. Some photos links or pages could then be shown or not depending on the user that is logged.

To set up in-page security, you need to use the `isEnabled($usersList,$groupsList)` function. Its functionment is exactly the same as the page level security `pageProtect($usersList,$groupsList)` function. You need to provide the `isEnabled()` function **two arrays** that specify which users and or groups can access the desired ressource.

Example:

```
<?php  
    $usersList = array("test_sample_3");  
    $groupsList = array("test");  
    if (isEnabled($usersList,$groupsList)) {  
?>
```

```
<h4>Section enabled</h4>
This section is enabled only for user 'test_sample_3' and users of the group
'test'.
<?php
    }
?>
```

The above example will display Section 4 only if the connected user is test_sample_3 or is a member of the group test.

5 Feedback, bugs and comments

If you noticed errors in this documentation, please inform by mail at [<vincentarn@users.sourceforge.net>](mailto:vincentarn@users.sourceforge.net).

6 Licence

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY

OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

7 News and updates

phpAuthent on Sourceforge.net

The phpAuthent website is accessible at <http://phpauth.sf.net>. Be aware that the project name (phpAuthent) and the website address differ.

phpAuthent is hosted on Sourceforge.net, the project page (downloads, support, mailing list, forum) can be accessed at <http://sourceforge.net/projects/phpauth/>.

A mailing list has been set up and is fed by any news release. Please sign-up to get latest news (very low number of messages a year). The mailing list subscription can be reached through the [project page](#).