

# Sitelok™



V5.4

## **Sitelok Manual**

Copyright 2004-2017 Vibralogix. All rights reserved.

This document is provided by Vibralogix for informational purposes only to licensed users of the Sitelok product and is provided on an 'as is' basis without any warranties expressed or implied.

Information in this document is subject to change without notice and does not represent a commitment on the part of Vibralogix. The software described in this document is provided under a license agreement. The software may be used only in accordance with the terms of that license agreement. It is against the law to copy or use the software except as specifically allowed in the license.

It is the users responsibility to ensure the suitability of Sitelok before using it. In no circumstances will Vibralogix be responsible for any loss or damage of data or programs as a result of using Sitelok. Your use of Sitelok implies acceptance of these terms.

# Contents

<b>Chapter 1 Introduction.....</b>	<b>9</b>
What is Sitelok?.....	9
Features.....	9
An overview of how Sitelok works.....	10
<b>Chapter 2 Installation.....</b>	<b>13</b>
Installing Sitelok.....	13
Installing for the first time.....	13
Upgrading an existing installation of Sitelok.....	14
Basic configuration.....	15
<b>Chapter 3 Example members area.....</b>	<b>18</b>
Registering a user.....	18
Logging in to the members area.....	19
Modify profile page.....	20
Logout link.....	21
Login Redirection.....	21
Login form on a public page.....	22
Changing the look of the default login template.....	23
Other examples.....	23
Sitelok Control Panel.....	23
Important: Keeping user passwords secure.....	24
<b>Chapter 4 Sitelok Control Panel.....</b>	<b>26</b>
Usernames.....	26
Passwords.....	26
Email addresses.....	26
Custom fields.....	27
Usergroups.....	27
Expiry dates.....	28
Accessing the control panel.....	28
Navigating the user list.....	28
Adding users.....	29
Editing users.....	31
Managing usergroups.....	31
Send email to a user.....	34
Delete a user.....	34
More options.....	34
Recent activity.....	34
Send password reminder.....	34
Filtering and finding records.....	34
Quick Filters.....	35
Advanced filter.....	36
Enter SQL query.....	37
Operations on multiple users.....	37
Select All users.....	37
Deselect all users.....	37
Refresh.....	38

Email selected users.....	38
Delete selected users.....	38
Export Selected users.....	38
Importing Users.....	39
The Sitelok Log.....	42
Email system.....	44
Email templates.....	45
HTML editor.....	46
Email template variables.....	47
Backing up.....	58
Plugins.....	59
Styling the default login template.....	59
Registration form code.....	60
Update profile form code.....	60
Login form code.....	60
Contact form code.....	60
SUBADMIN access.....	61
Translating the control panel.....	62
<b>Chapter 5 Securing Pages.....</b>	<b>64</b>
Allowing access to certain usergroups.....	64
Allowing access to certain users only.....	65
Adding a logout link.....	65
Securing download links.....	66
<b>Chapter 6 Displaying user data in pages.....</b>	<b>70</b>
\$slusername.....	70
\$slpassword.....	70
\$slpasswordclue.....	70
\$slname.....	70
\$slfirstname.....	70
\$sllastname.....	71
\$slemail.....	71
\$slcustom.....	71
\$slstarttime.....	71
\$slaccesstime.....	71
\$slcreated.....	71
\$slusergroups.....	72
\$slgroupname, \$slgroupdesc and \$slgroupexpiry.....	72
\$slpublicaccess.....	73
\$sljustloggedin.....	73
Force database read on page access.....	73
<b>Chapter 7 User Registration Page.....</b>	<b>75</b>
Creating a registration form using the Form Designer.....	76
Sending registration data to the thank you page.....	79
Auto login after registration.....	80
Admin approval of users before access.....	80
User confirmation of email address before access.....	82
Creating a registration form using your own html.....	84
<b>Chapter 8 User Profile page.....</b>	<b>91</b>

Creating an update profile form using the Form Designer .....	91
Requiring existing password to make changes.....	94
Requiring confirmation of email change.....	94
Creating a profile update form using your own html.....	95
<b>Chapter 9 Login Template .....</b>	<b>103</b>
Login page types.....	103
The default Login Template.....	103
Creating a login template using the Form Designer.....	104
Creating a login template using your own html form.....	106
<b>Chapter 10 Login Pages .....</b>	<b>111</b>
Login page types.....	111
Creating a login form using the Form Designer .....	111
Creating login form using your own html.....	113
<b>Chapter 11 Contact Forms .....</b>	<b>118</b>
Creating a contact form using the Form Designer .....	118
Sending contact form data to the thank you page.....	122
<b>Chapter 12 User messages &amp; warnings.....</b>	<b>123</b>
Message and error page.....	123
Expired membership page URL.....	123
Wrong group page.....	124
No access page URL.....	124
Forgotten password email template.....	125
Translating or modifying messages .....	125
<b>Chapter 13 Public access and page content control.....</b>	<b>127</b>
Allow public access to a page.....	127
Display section of page only for public visitors.....	128
Display section of page only for members .....	128
Display section of page for members of specific group.....	128
Display section of page for active members of specific group.....	129
Display section of page for specific user.....	129
Login link and redirect back to this page.....	129
Download links for public visitors .....	129
Allow user to join a new usergroup.....	131
Allow user to remove themselves from a usergroup.....	131
Allow user to replace a usergroup with a new one.....	132
Allow user to extend a group membership.....	133
Allow user to delete their account.....	134
Send email to user .....	134
<b>Chapter 14 Search engine access.....</b>	<b>135</b>
Enable search engines to index pages.....	135
Overriding the setting for individual pages.....	135
The search engine user .....	136
Verifying search engines are who they say.....	136
<b>Chapter 15 Configuration.....</b>	<b>138</b>
General Settings .....	138
Site name .....	138
Admin email address.....	138

Date format.....	138
Site Key.....	138
Maximum session time.....	138
Maximum inactivity time.....	139
Login CAPTCHA.....	139
Register CAPTCHA.....	139
Remember me (cookie login).....	139
Allow concurrent logins.....	139
Force database read on page access.....	140
Allow search engine access to protected pages.....	140
Search engine usergroup.....	140
Email change verification.....	140
Email field must be unique.....	141
Login with username or email.....	141
Passwords.....	141
Forgotten password email template.....	141
Profile update password required.....	142
Store password as hashes.....	143
Pages & Templates.....	144
Logout page URL.....	144
Message and error template.....	144
Login template.....	144
Expired membership page URL.....	145
Wrong group page.....	145
No access page URL.....	146
Log Settings.....	146
Log entry types.....	146
Server path to text format log file.....	147
Custom fields.....	147
Custom X field title.....	147
Custom X size.....	147
Custom X Validation.....	147
Dashboard.....	148
Main table column order.....	148
Default rows to display in admin page.....	148
Main table font size.....	148
User page layout.....	148
Download paths.....	148
Full file path to download folder.....	148
Additional download folders.....	149
Email settings.....	149
Send email using.....	149
Reply to email override.....	150
Allow from name and reply-to override on send email dialog.....	150
Send copy of emails sent from dashboard.....	150
Delay between emails during bulk delivery.....	150
Send test email.....	150
Install paths.....	151
Full file path to the Sitelok folder.....	151
URL to Sitelok folder.....	151

Full file path to email template folder .....	151
URL to email template folder.....	151
Full file path to backup folder .....	152

**Chapter 16 Sitelok API.....153**

slapi_getuser.....	153
slapi_adduser.....	153
slapi_modifyuser.....	154
slapi_deleteuser.....	154
sl_delete.....	155
slapi_addorupdate.....	155
slapi_getallusernames.....	157
slapi_getallusergroups.....	157
slapi_addgroup.....	158
sl_addgroup.....	158
slapi_removegroup.....	158
sl_removegroup.....	159
slapi_replacegroup.....	159
sl_replacegroup.....	160
slapi_extendgroup.....	160
sl_extendgroup.....	161
sl_ismemberof().....	161
sl_isactivememberof().....	161
sl_memberofexpires().....	161
sl_ismemberofany().....	162
sl_ismemberofall().....	162
sl_isactivememberofany().....	162
sl_isactivememberofall().....	162
sl_setcustom().....	163
sl_redirectourl().....	163
sl_getstartpage().....	163
slapi_loginuser.....	163
sl_logout().....	164
sl_siteloklink().....	164
slapi_totalusers().....	164
slapi_usergrouptotalusers().....	164

**Chapter 17 Sitelok event handler.....165**

sl_onCustom1Validate.....	165
sl_onUsernameValidate.....	166
sl_onPasswordValidate.....	166
sl_onEmailValidate.....	167
sl_onNameValidate.....	167
sl_onAddUser.....	168
sl_onModifyUser.....	168
sl_onDeleteUser.....	169
sl_onLogin.....	169
sl_onLogout.....	169
sl_onDownload.....	170
sl_onDownloadEmail.....	171
sl_onCheckLogin.....	171

sl_onCheckRegister .....	172
sl_onCheckModifyProfile.....	172
sl_onPageAccess.....	173
sl_onAddGroup .....	173
sl_onModifyGroup .....	174
sl_onDeleteGroup .....	174
sl_onPaypalOrder .....	174
sl_onUpload .....	175
sl_onPrepareEmail.....	175
sl_onSendEmailAllowed.....	175
sl_onSendEmailOut.....	175
sl_onLoginFailure.....	176

**Chapter 18 Support.....177**

# Chapter 1 Introduction

## What is Sitelok?

Sitelok is a PHP and MySQL session based application that allows you to password protect and control access to html\* or PHP pages within your website. You can separate your site into multiple members only areas and each user can be a member of one or more areas. Depending on your requirements new users can be added to Sitelok either manually via the Sitelok control panel, via payment plugins, automatically from a PHP script using the Sitelok API or via a registration form that allows a user to add themselves. A users membership of an area can be set to expire after a fixed number of days or can be left open. Once setup securing a page is simply a matter of adding a simple code snippet to the top of the page. The Sitelok control panel provides an easy to use and powerful way to manage users and check access via the log features. The control panel also has a powerful template based email system allowing you to keep in touch with your users.

\*To secure .html pages you should set your server to allow PHP to be used inside .html pages. This is usually a simple .htaccess setting.

## Features

### General

- Uses PHP and MySQL available on most servers.
- Easy to setup and maintain.
- Protects access to PHP and HTML pages (just rename html pages with the php extension).
- Unlimited users and secure areas (usergroups).
- Users can be members of multiple groups.
- Powerful administration control panel.
- Fully compatible with SSL.
- Can store passwords either in plain text or secure hashes (recommended). Sitelok uses PHP's built in password hashing system which takes the user's password and hashes it using the current best algorithm (bcrypt, as of writing) and a securely created salt. This is an extremely secure way to protect users passwords from hackers.

### Protected areas

- Fully customizable login page is displayed whenever a user attempts to access a page that is protected.

- Login form has a 'forgot my password' feature.
- Optional 'remember me' and auto login features
- Unlike other systems Sitelok can also protect resources within the pages such as graphics and download files using our built in Linklok secure link system.
- Membership of each group can be set to expire individually for each user.
- Fully customizable registration form allows new users to register. Ideal for free membership areas.
- Registration form can either allow users to set their own login details or can assign passwords etc automatically. There are many options for admin approval or user confirmation.
- The Sitelok API can be added easily to PHP scripts (such as Linklok) allowing users to be added automatically. Email templates can be used to inform the user and administrator of the new or modified user details. Ideal for use with paid membership sites.
- Modify profile form allows users to change their registered details.
- User defined pages displayed at logout or when a users membership has expired.
- Ability to use and display within protected pages the username, name, password, groups and expiry times and custom fields applicable for the current user.
- Pages can be defined as public allowing non logged in users access to all or parts of a page. You can completely control what public and logged in users see on pages.
- Option to allow search engines to index protected pages.

### **Administration control panel**

- Add, edit, delete or temporarily block individual users.
- Email, delete, import or export individual or groups of users
- Searching and filtering of users.
- Ability to send template based email to one or multiple users selected on any criteria.
- Wysiwyg html email editor.
- View or export log of logins and logouts, failed login attempts, downloads of protected files, emails sent and other details including the users IP address.
- For advanced users a direct SQL query can be executed.
- Easy to use backup system for MySQL tables.
- Plugin system
- Form designers to help create registration, update profile, login forms and contact forms.
- Responsive design works well on mobile devices.

### **An overview of how Sitelok works**

When a user attempts to directly access a protected page Sitelok will display the login form (login template). The user enters their login details and if the details are authenticated against the user details stored in the database a PHP session is started allowing the user to access all the

protected pages in each group that he is a member of. If a user forgets his password then an automated system will send a reminder email to the registered email address. Access is allowed until logout is clicked, the browser is closed or the session times out. Use the built in login form designer or provide your own html login form. Either way Sitelok will fit seamlessly into your site. Login forms can redirect users to a page based on usergroup.

Once a user has logged in he can navigate through the permitted secure pages as normal. Within these pages you can embed the users details such as name, email address, username, group names, expiry times and custom data from the database. Unlike most non htaccess based systems Sitelok allows you to protect not only the pages themselves but also embedded files and links such as graphics and downloadable files using our Linklok system to ensure that the links only work while a user is logged in. If a user attempts to access a page belonging to a group to which he is not a member then a customized error page is displayed. Likewise if a users membership expires he will be taken to a page of your choice. Modify profile forms allow users to change their details such as password, name or email address or can be used to collect further information from them. Finally when a user is finished he can use the logout function to end the session. The Sitelok database can also store up to 50 custom fields of information for each user.

Sitelok provides four ways to add new users (or extra group memberships for existing users) depending on the type of membership you offer on your site.

- Users can be added manually using the Sitelok control panel.
- Users can add themselves using a customizable registration form. The user can define their own username and password if required or Sitelok can assign a password and email it to the user. There are options for a CAPTCHA field and to have the administrator approve registration before access is allowed or have users confirm their email to activate their account.
- You can use a Sitelok payment plugin (such as Paypal, Stripe or 2CO) to sell membership either as a one off purchase or a subscription. User accounts are created and updated automatically when payment is received.

- You can add the ability to other PHP scripts (such as Linklok) to add and modify users by using the functions provided in the Sitelok API. This allows you to charge for membership and automatically add the user and email the login details.

Sitelok has an easy to use control panel that allows you to add and modify users, send emails, import and export user details, search and filter users, perform SQL queries, manage the access log and many other features.

# Chapter 2 Installation

## Installing Sitelok

Before installing Sitelok make sure that your server has installed PHP 5.3.7 or above and MySQL. You will also require an existing MySQL database and user before continuing.

So that you can get up and running quickly we will start by setting Sitelok up using mostly default settings. Once everything is working we will describe how you can customize the system and explain the various features that you can make use of.

## Installing for the first time

**Important.** Some editors try to use smart quotes (that look curly or angled) instead of the simple " character. These look great in print but don't work on webpages. If you get a blank white page with no errors shown when you try to install then this is quite likely the cause. The editor should allow you to disable smart quotes.



If you are upgrading an existing installation please skip this section and go directly to the section about upgrading.

- 1) Extract the contents of the sitelokpw.zip file to your PC and find the file called slconfig.php in the slpw folder.
- 2) Open slconfig.php in a text editor (for example notepad on Windows or TextEdit on a Mac) and you will see the following settings (there may be other settings which you can ignore).

```
$DbHost="localhost"; // Database host
$DbName="dbname"; // Database name
$DbUser="username"; // Database username
$DbPassword="password"; // Database user password
```

Set these to the MySQL details for your server. Your hosting company should be able to provide this information if you are unsure. Just enter the settings between the " characters (the parts in bold).

- 3) Save the slconfig.php file with your settings.

- 4) Upload the slpw and members folders to the root of your site using FTP. There are no special permissions required on most servers.
- 5) Open your browser and go to <http://www.yoursite.com/slpw/install.php> which will start the installation process. If you see any errors please double check your MySQL settings in slconfig.php and try again. If all is well you will see a message saying 'Installation completed successfully'. On some servers install may not have permission to create some required folders in the root of your site or to set write permission on the slpw/email folder. In this case you will be asked to do this manually using your FTP client.
- 6) **Important.** After a successful installation you should delete the [install.php](#) and [upgrade.php](#) files from the slpw folder on your server.

If you have any problems with installation please let us know so that we can help you.

### Upgrading an existing installation of Sitelok

If you have just installed Sitelok for the first time you can ignore this section.

**Important.** Make sure your server supports PHP 5.3.7 or above and has the mysqli library included (should be by default). Also make sure all plugins are updated to the latest version before upgrading.

Before continuing make a backup of your existing slpw folder in case of problems.

- 1) Extract the contents of the sitelokpw.zip file to your PC.
- 2) Upload and overwrite all of the files in the slpw folder (and subfolders) **except for**
  - [slconfig.php](#)
  - [email](#) folder
  - [eventhandler.php](#) if you have made custom changes to it
  - [writefiles](#) folder if you are upgrading from V5.0 or above
  - [emailthumbs](#) folder if you are upgrading from V5.0 or above
- 3) Open your browser and go to <http://www.yoursite.com/slpw/upgrade.php> which will start the upgrade process. If all is well you will see a message

saying 'The upgrade was successful'.

- 4) If you are upgrading from **V1.43 or earlier** then open your existing `slconfig.php` file and add these lines near to your other MySQL settings.

```
$DbLogTableName="log";  
$DbConfigTableName="slconfig";  
$DbGroupTableName="usergroups";
```

Upload the modified `slconfig.php` to the `slpw` folder.

Now that you have upgraded the settings you originally had in `slconfig.php` will have been copied to the new mysql table based configuration. Any changes need to be made in the control panel rather than by editing the `slconfig.php` which we are sure you will find easier. `slconfig.php` is used only for mysql settings now. Logging of access is also now stored in a mysql table but your upgrade will leave enabled the text based log file as well. You can select which to use in the configuration.

- 5) **Important.** After a successful upgrade you should delete the `install.php` and `upgrade.php` files from the `slpw` folder on your server.

### Basic configuration

Before using Sitelok for the first time we should configure some basic details such as your admin email address etc.

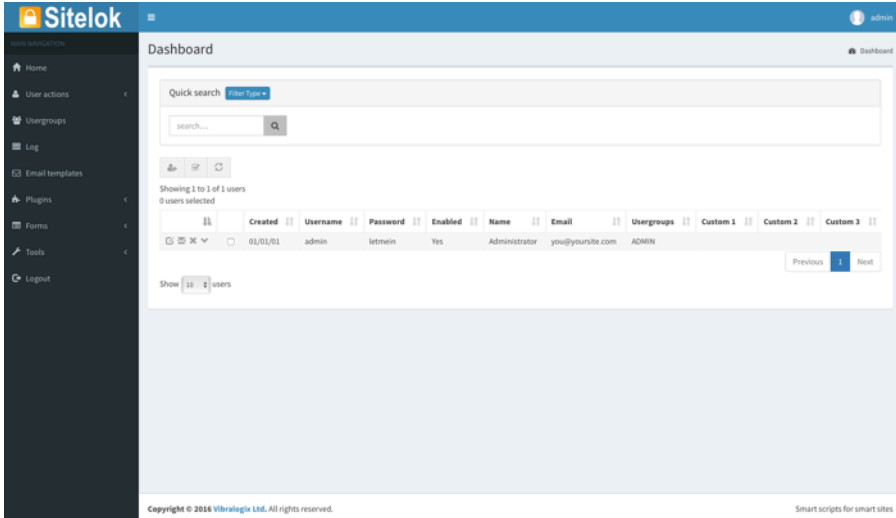
Login to the Sitelok control panel at

<http://www.yoursite.com/slpw/>

Username: `admin`

Password: `letmein`

You should see the control panel main screen appear.



Click the **Tools** menu on the right and select **Configuration** to display the setup page.

Most of the settings can be left at their default values but a few should be set now before continuing. In the **General Settings** panel

**Site name** Set a name for your members area

**Admin email address** Enter the site administrators email address. This should be on the same domain as your website ideally.

**Date format** This is the date format used by Sitolok for expiry dates. Once you set this format you should not change it again. Choose either DDMMYY (european format) or MMDDYY (US format).

<b>General settings</b>	General settings
<b>Passwords</b>	
<b>Pages &amp; templates</b>	
<b>Log settings</b>	
<b>Control panel</b>	
	<b>Site name</b>
	<input type="text" value="Members Area"/>
	<b>Admin email</b>
	<input type="text" value="you@yoursite.com"/>

Scroll down and click the [Update general settings](#) button.

In the [Pages and Templates](#) panel

**Logout page URL** Enter the full URL to the page users should be sent to after they logout. You can override this from individual pages later if you wish.

General settings	Pages & templates
Passwords	<b>Logout page URL</b>
<b>Pages &amp; templates</b>	<input type="text" value="http://www.yoursite.com"/>

Scroll down and click the [Update template settings](#) button to save the changes.

You will learn about the other settings you saw in the configuration page later in this manual as each feature is described.

You can logout of the Sitelok control panel now if you wish.

## Chapter 3 Example members area

So that you can get started quickly we have included a simple example members area that includes a user registration form, a couple of member only pages and a modify profile page.

You can use these pages as a tutorial and modify them for your own use if you wish. They are intentionally plain and simple so that you can see easily how they work.

The members folder is just for the examples but your own members pages can be anywhere on your site. Also don't forget to delete the example members folder or any pages in it that you don't use yourself after you have finished.

**Important.** Before continuing make sure you have logged out of the Sitelok control panel. This avoids any confusion that can occur when you pretend to be different users all on the same PC!

### Registering a user

In our example members area we will allow users to signup for membership themselves by filling in a registration form. We will ask for their name and email address which is the minimum required information and also some extra information that we will store in two of the custom fields (there are 50 custom fields you can use). We will use the email address as the username and have Sitelok generate a password and send it to the email address supplied. Using this method ensures that the entered email address really exists and that the user has access to it. Of course Sitelok also allows you to create registration forms where the user chooses their own username and password if you prefer.

To see the registration page go to

<http://www.yoursite.com/members/register.php>


If you have enabled CAPTCHA for registration forms in the Sitelok configuration then you should use the example called `registerturing.php` instead of `register.php`.

## Chapter 3 - Example members area

---


To test it enter your name (use a first and last name), your real email address and any data you want in the example custom fields.

When you click the register button Sitelok will create a new Sitelok user. They will be a member of the usergroup called CLIENT with no expiry set. An email will be sent to the email address entered and also to the Sitelok administrator.

You can edit the style and features of the registration form on register.php in the control panel. Just go to **Forms - Registration forms** and click  next to **Example for register.php**.

     Example for register.php

As the form was embedded in the page any changes you make to the form in the designer and save will appear on the page immediately.

If you want to add a registration form to your own page you can design the form either by editing the example or by creating a new one. Click the  button to generate the code to paste in your page.

As well as embedding a form in the page you can generate the full html, css and javascript which you can modify to your requirements before pasting in the page. Its also possible to use your own form code as well.

See the chapter about Registration Pages later in the manual for more details about creating registration forms.

### Logging in to the members area

Now that you have registered for the members area you can try to view the protected pages. In your browser click the link displayed in the registration thank you page or go directly to.

<http://www.yoursite.com/members/members.php>

Notice that because this page has been protected by Sitelok it will display a login form. This login form is displayed whenever a user tries to access a protected page when they are not yet logged in.

If the login form does not appear then probably you forgot to logout of the control panel previously (i.e. you are already logged in). Users who are members of the ADMIN group can see all protected pages.

Enter the username (your email address) and password that were emailed to you when you registered and you should then see the members only page displayed. As you can see you can display user specific data like the users name, email address etc on members pages. In our example we also have a logout link, a link to the modify profile page and a link to another members page.

You can adjust the style of the default login form in the admin control panel in **Forms - Default login form style**. You can also replace it entirely with your own design as described later in the manual.

If you open up members.php you will see that protecting it just required the following snippet to be added to the top of the page.

```
<?php
$groupswithaccess="CLIENT";
require_once("../slpw/sitelokpw.php");
?>
```

These lines tell Sitelok to control access to the page and only allow access to members of the usergroup called CLIENT.





### Modify profile page

If you click the modify profile page link or go directly to


<http://www.yoursite.com/members/update.php>

you will see a form that allows users to changes their details such as password, name, email address, custom fields etc. As with the registration form you have complete control over the form design and which data can and can't be changed by the user. Also as with the registration form you can define email templates to be sent to the user and admin when the form is submitted.

You can edit the style and features of the update form on update.php in the control panel. Just go to **Tools - Update form code** and click  next to **Example for update.php**.

    Example for update.php

As the form was embedded in the page any changes you make to the form in the designer and save will appear on the page immediately.

If you want to add an update form to your own page you can design the form either by editing the example or by creating a new one. Click the  button to generate the code to paste in your page.

As well as embedding a form in the page you can generate the full html, css and javascript which you can modify to your requirements before pasting in the page. Its also possible to use your own form code as well.

See the chapter about User Profile Pages later in the manual for more details about creating the forms.

### Logout link

The final part of our very simple members area is the logout link. If you click this it will end the users login session and display the page set in the Logout Page setting in the configuration. You can also override the default logout page for any logout links as well if you need to.

### Login Redirection

As you have seen when you try to visit a secured page (and you are not logged in) Sitelok displays the login template. This is how most membership systems work and is useful if users bookmark pages of your site for direct access. However you probably want to have login links or buttons on your site that the user can click to login and then be redirected to a suitable page for the usergroup they belong to. To do this you can point the login button or link to </slpw/login.php>. To try this go to

<http://www.yoursite.com/slpw/login.php>

and login as the member you registered as. You should be redirected to `/members/members.php` after login. This redirect URL has been set for the usergroup CLIENT in the **Usergroups** page in the Sitelok control panel and can be changed as needed. If you had logged in as the admin user you would have been redirected to the admin control panel (the url set for the usergroup ADMIN).

Pointing to `/slpw/login.php` is an easy way to add Login links to your site.


## Login form on a public page

As well as simply pointing a Login button to login.php SiteLok also allows you to have login forms on pages of your site where users can choose if they want to login or not. SiteLok can also control public pages of your site so that the information displayed can differ between what a public users sees and what a logged in member sees. The information shown can even be controlled based on the group the user belongs to or even the specific user.

To see a simple example open


<http://www.yoursite.com/members/pagewithlogin.php>

If you are not logged in you will see a page with the login form and other parts that can be seen by a public visitor. If you login then the user will be redirected to the start page for their usergroup, if one is set. If pagewithlogin.php is visited again while already logged in it will appear different. The login form is hidden and a logout link is displayed instead. The logout link on this page has been set to call the same page when clicked rather than to display the default logout page.

You can edit the style and features of the login form on pagewithlogin.php in the control panel. Just go to **Forms - Login forms** and click  next to **Example for pagewithlogin.php**.

     Example for pagewithlogin.php

As the form was embedded in the page any changes you make to the form in the designer and save will appear on the page immediately.

If you want to add a login form to your own page you can design the form either by editing the example or by creating a new one. Click the  button to generate the code to paste in your page.

As well as embedding a form in the page you can generate the full html, css and javascript which you can modify to your requirements before pasting in the page. Its also possible to use your own form code as well.

If you open up the page in your editor you will see that we have used some very simply code to determine what gets included in the page at different times. The details about this will be explained later in the manual.

For more information about login pages see the chapter called Login Pages later in the manual.

### Changing the look of the default login template

You can replace the default login template displayed when you try to login with your own so that it matches your site perfectly. However in many cases the default form may just need some styling to match your site. In the admin control panel you can edit the form by going to [Forms - Default login form style](#). Here you can change colors, fonts and the text of the form. You can also add a background color or image for the page. You may wish to add a link to your registration form (or the page where users pay to join) that can be clicked by users who don't have a username or password yet.

The login form has been designed to be responsive so that it works on mobile devices as well.

### Other examples

You will also see the following example files in the members folder but we recommend looking at these when you get to the relevant part of the manual.

<a href="#">download.php</a>	Example Sitelok download link
<a href="#">registerturing.php</a>	Registration form with CAPTCHA code
<a href="#">registerapprove.php</a>	Registration form requiring admin approval
<a href="#">contactform.php</a>	Example contact form

As well as the examples in the folder members there is a folder called [html source examples](#) which contains example forms using html code rather than embedded forms. Its best to use these examples as references to the relevant parts of the manual. In most cases the form designers will create any forms that you will need.

### Sitelok Control Panel

Now you have used Sitelok as an end user would in the example members area we will take a look at the admin control panel.

**Important.** Before continuing make sure you logout from the example members area.

Login in to the Sitelok control panel at

<http://www.yoursite.com/slpw/>

Username: [admin](#)

Password: [letmein](#)

You should now see two users in the database. The first is the admin user and the second is the user you registered using the registration form. Notice that the username you registered is a member of the usergroup called CLIENT and has an expiry date one year from now.

Most of the control panel is straight forward and doesn't need too much explanation at this point. You will see that you can add new users, edit existing users, view the access log and email users. Try some of the options to familiarize yourself with how things work. Each feature is described in full later in the manual of course anyway.

### **Important:** Keeping user passwords secure

The default installation of Sitelok stores passwords in plain text which allows you to see them in the control panel and the forgotten password option can send out the password as a reminder. This makes the tutorial easier to follow. However we strongly recommend for live sites that you enable Password hashing. To do this

- 1) Go to [Tools - Configuration - Passwords](#) and enable [Store passwords as hashes](#).

Store passwords as hashes

**Important** See the manual for more details

- 2) Change the default forgotten password email template to be [forgotpasshash.htm](#).

Forgotten password email

[forgotpasshash.htm](#) 

Instead of showing the users password this email provides a new random password which the user can enable. They can of course change this password using a modify profile page once they are logged in, if you allow this.

Once enable it means that if a hacker were to get access to MySQL on your server he will not see your users passwords which may of course be the same passwords they use on other sites.

## Chapter 4 Sitelok Control Panel

The Sitelok control panel provides complete control over your user database. Although simple to use it has a powerful user interface that allows you to add new users, edit existing users, block users, modify expiry dates, send email, view complete access logs and many other features. One very useful feature is the ability to select users based on any criteria and to send to that group template based email. Sitelok's built in wysiwyg html email editor makes this task very simple.

The control panel supports most modern browsers and is responsive to work well on mobile devices too.

**Important.** The control panel requires javascript to be enabled in your browser (this is enabled by default in most browsers anyway).

Before we try out the various parts of the control panel we will explain a few concepts used by Sitelok.

### Username

A members username must be unique to each user in the database. You can use any alphanumeric characters and the symbols @-\_. in usernames. Often it is a good idea to use an email address for a username as this is easy for a user to remember and is also unique. Usernames have a maximum length of 100 characters.

### Passwords

Passwords can contain any alphanumeric characters. When adding users you can have Sitelok generate a random password if you wish. By default this will consist of 4 letters followed by 2 numbers (e.g. tyhg95). You can change the style of random passwords in the 'Random Password Mask' setting of the configuration page. Passwords have a maximum length of 50 characters.

### Email addresses

The email field of a users entry should contain their email address. Even if you have user email addresses for the username it is still this email address that is used for communication. Maximum length of email addresses is 100 characters.

## Custom fields

Sitelok has up to 50 custom fields you can use to store information about users. Custom fields can be used to display data in members pages or can be used only for your administration use. They can also be used as criteria when selecting users in the control panel for emailing etc.

The control panel will only display custom fields that have been assigned a title in the configuration page. This avoids messing up the display with unused custom fields. By default only the first ten custom fields have titles when you first install Sitelok but you can rename these or add new titles to other custom fields.

The maximum length of each custom field is by default 255 characters. If you require more characters to be stored you can set the size of each custom field in [Tools - Configuration - Custom fields](#).

## Usergroups

When you protect a page on your site you will normally define which usergroups have access to that page. Only users that belong to one of the allowed groups will be able to view the page (ADMIN users can access any page). Users can be members of any number of usergroups and you can optionally define for each group an expiry date for that user.

You can use alphanumeric characters and the symbols #{}().@. in user group names. The usergroup ADMIN is a special existing group name whose members have access to the Sitelok control panel and all other secured pages. The ALL usergroup also allows access to all secured pages but not the control panel. There is also a special usergroup called SUBADMIN which allows users access to the admin panel but not to access things like the configuration settings.

Usergroups in Sitelok do not have to be defined in advance. You can simply create a page that allows for example a group called GOLD and then create a user and enter GOLD in their usergroup field. This allows you to have unlimited groups and is also great for developers who use Sitelok as part of other systems.

## Expiry dates

Each usergroup that a member belongs to can also have an optional expiry date. This is entered in the format selected in the configuration page. This will be DDMMYY or MMDDYY. You should not change this settings once you have started adding users to Sitelok. This same date format is used wherever you need to use a date in Sitelok.

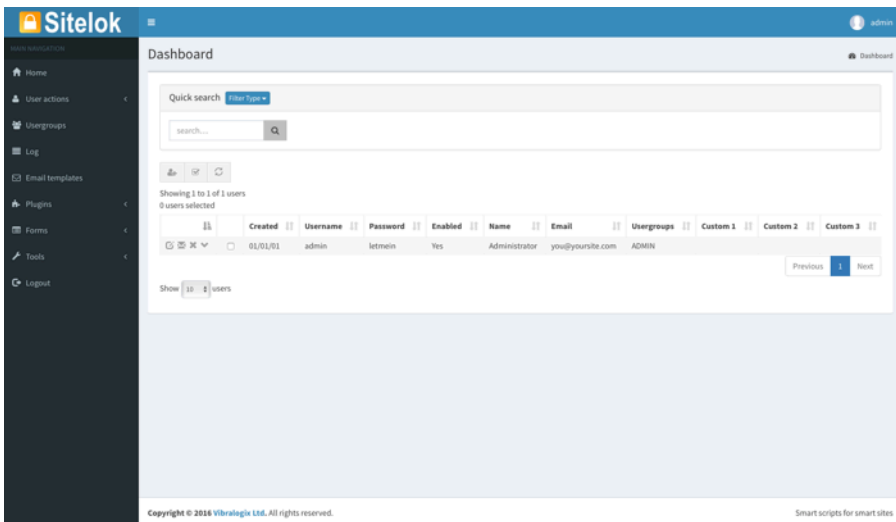
## Accessing the control panel

To access the control panel go to

<http://www.yoursite.com/slpw/>

Username: **admin**

Password: **letmein**



The main screen of the control panel displays a list of all members.


## Navigating the user list

At the moment you probably only have a few users added to Sitelok but hopefully soon you will have many more! You can navigate between pages of users using the navigation controls



You can also change how many users are displayed per page. You can sort the user list by clicking the column headings. Clicking again will reverse the order. To cancel sorting and revert to the order in which mysql has the user records stored click the blank column heading above the first table column.

## Adding users

To add a new user to Sitelok select **Add User** in the **User actions** menu or click the  button above the user table.

A screenshot of the 'Add User' form in the Sitelok Control Panel. The form is divided into several sections: 'User Details' with fields for Username, Password (with a 'Random' button), Name, and Email, and a checked 'Enabled' checkbox; 'Usergroups' with a dropdown menu, a 'days or DDMMYY' field, and a '+', '-', and 'x' icon; 'Other Options' with a checked 'Email user on save' checkbox and a 'newuser.htm' link; and 'Custom Fields' with ten empty input boxes labeled 'Custom1' through 'Custom10'. At the bottom, there are 'Add User' and 'Return to dashboard' buttons.

Most of the add user form is straightforward asking for a unique username, password, name, email address, usergroups and any custom fields you have defined. By default only the first 10 custom fields are displayed but others will be shown if you define titles for them in the configuration.

## Chapter 4 - Sitelok Control Panel



---

The usergroup entries should contain the usergroup name that the user should belong to. This will decide which pages they can access on your site. So for example if you have added this to the top of a page

```
<?php
$groupswithaccess="CLIENT";
require_once("../slpw/sitelokpw.php");
?>
```




and you want this user to have access then you need to enter CLIENT as one of their usergroups. You can either type the usergroup name directly or select an existing group from the drop down menu. The drop down menu will only list usergroups that you have predefined in the Manage User Groups page.


Although the usergroup interface looks like a normal html drop down menu you can in fact type directly into the text box as well. This is like a software applications combo box.

If you want to limit membership only until a certain date then you should also enter an expiry date next to the usergroup name. You can click the  icon to select a date from the calendar or you can type the date directly in the format DDMMYY or MMDDYY depending on what you defined in the configuration page. Alternatively you can type in a number of days until expiry (such as 365 for a year). If you don't need an expiry date just leave it blank. Use the  to add further usergroups.

Here is an example

### Usergroups

CLIENT	days or DDMMYY		▲ ▼ ✕
GOLD	240817		▲ ▼ ✕
SILVER	30		▲ ▼ ✕




Here we have assigned the user to three usergroups. They have unlimited access to the CLIENT usergroup, access until 24th August 2017 for the GOLD usergroup and 30 days of access for the SILVER usergroup.

- ✘ Next to each usergroup entry you will see a delete icon you can use to delete the group.
- ▲ ▼ The two arrow icons allow you to rearrange the order of the usergroups. The order of the usergroups is normally of no importance to Sitelok. The only exception is if you define a usergroup that redirects a user upon login in which case the first group listed (that has not expired) will be used.

Select the **Email user on save** checkbox and select an email template (see later) if you wish Sitelok to send an email to the user after adding them to the database.

### Editing users

To edit an existing user click the edit icon  next to the user in the main table. You can then change any of the users settings in the same way as the Add User form. If you change the username Sitelok will ensure that it is unique (not already in the database). When you make changes to a users account if the user is currently logged in they will only see the changes when they next login. One exception to this is if you disable a user as this will destroy their existing session to block access immediately.

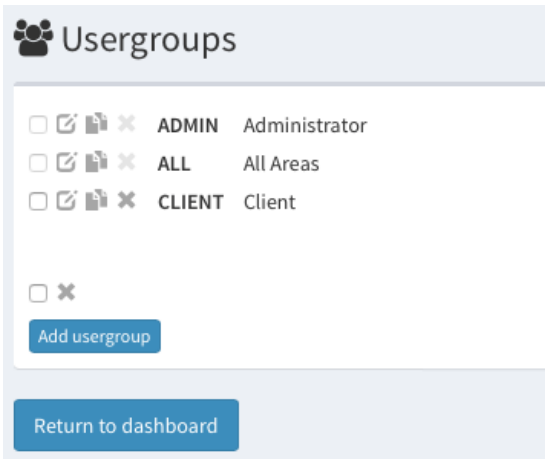
if you have set Sitelok to not allow concurrent logins (see the configuration section) you can also force logout a user by selecting the **Force logout on save** checkbox before saving changes.

Select the **Email user on save** checkbox and select an email template (see later) if you wish Sitelok to send an email to the user after adding them to the database.


### Managing usergroups


As we mentioned earlier user groups do not need to be defined as such in Sitelok. When you create a user you can enter any group names you wish as you can when you create secured pages and select which groups have access. Sitelok does not need to know anything about the groups to function properly. However Sitelok does let you create usergroups in advance if you wish. These will appear in drop down menus in the control


panel which makes things easier and less prone to errors. To access the usergroups click **Usergroups** in the main menu.

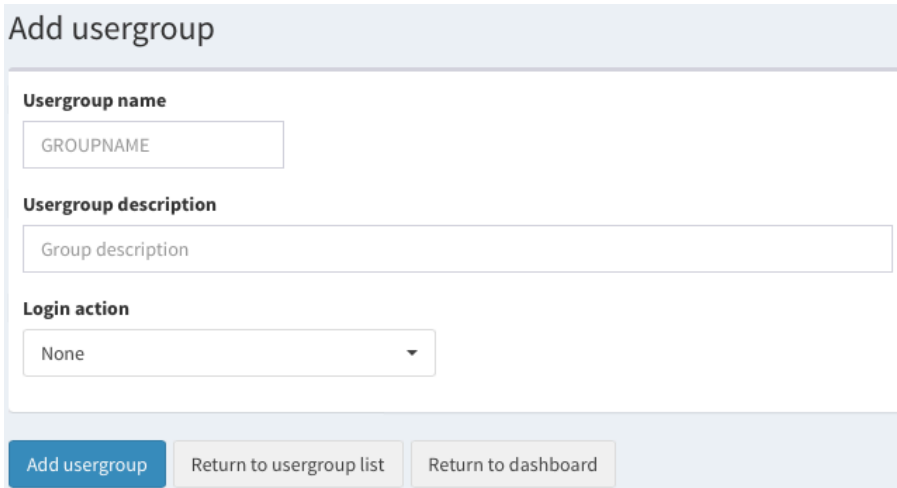


The screenshot shows the 'Usergroups' management page. At the top left is a header with a group of people icon and the text 'Usergroups'. Below this is a list of user groups, each with a checkbox, an edit icon (pencil), a delete icon (X), and the group name and role. The groups listed are: ADMIN (Administrator), ALL (All Areas), and CLIENT (Client). There is an additional empty row with a checkbox and a delete icon. Below the list is a blue button labeled 'Add usergroup'. At the bottom of the page is another blue button labeled 'Return to dashboard'.

Click  to edit the user group

Click  to delete the user group. This only deletes the entry in the usergroup table and will not remove the usergroup from users that belong to this group who will still have access.

Click  to add a new group.



The screenshot shows the 'Add usergroup' form. The title 'Add usergroup' is at the top left. The form contains three sections: 'Usergroup name' with a text input field containing 'GROUPNAME'; 'Usergroup description' with a larger text area containing 'Group description'; and 'Login action' with a dropdown menu currently set to 'None'. At the bottom of the form are three buttons: a blue 'Add usergroup' button, a grey 'Return to usergroup list' button, and a grey 'Return to dashboard' button.

### Usergroup name

This is the name used within Sitelok to refer to this group. Examples are ADMIN, GOLD, SILVER etc.

### Usegroup Description

This is the group description that is displayed in email templates or secure pages using Sitelok variables described later.

### Login action

If you use login.php or have a login form on your website pages then you can choose to have users redirect to a certain page upon login. URL's should be fully specified such as <http://www.yoursite.com/page.php>. Note that if a user belongs to more than one usergroup then the first group listed in the user record will be used for login redirection.

#### Redirect to URL


Select this option to redirect the user upon login to the specified URL. This is useful to send users to their usergroups home page.

#### Redirect to entry - Custom X


Select this option to redirect the user upon login to the URL stored in the custom field in the users record. This is a powerful feature enabling user based sites (as opposed to just group based sites) to be created. You can choose which of the 50 supported custom fields to use.

Note that login redirection only works when using login.php or from a login page. If you login using the login template which appears when you directly access a secured page then no redirection takes place as its assumed this is the page you wanted to view. See elsewhere in the manual for a better explanation about the difference between login pages and logins templates.


## Send email to a user

To email an existing user click the email icon  next to the user in the main table. You have the option to create the email from scratch or to send a template based email and you can use either the text or html formats. We will describe how to use the email templates and functions in further detail later in the manual.


## Delete a user

Clicking the delete icon  next to a user in the main table will permanently remove the user from Sitelok including relevant log entries. If the user is currently logged in then their session will be destroyed to block further access.


## More options

If you click the more options icon  next to a user you will see a drop down menu with other options relating to the user. This can include options for plugins that you may install.

## Recent activity

Clicking the Recent activity icon  (inside the more [options menu](#)) will display this users most recent activity from the log (latest 25 entries). You can see the same thing directly by viewing the log but this icon provides a more convenient way to do this.


## Send password reminder

Clicking the  icon will send a password reminder email to the user. This will use the default password reminder email template set in the configuration page.


## Filtering and finding records

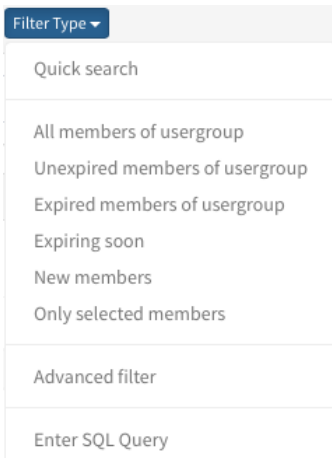
The Sitelok control panel has powerful set of filter and search options. Filtering users basically controls which users are listed within the table displayed on the main dashboard page. Once you have the users listed based on your filter criteria you can select some or all of them to perform other operations such as bulk email. Click the [Clear filters](#) link to display all the users again. Clearing filters will keep users selected so you can in fact

combine filters. For example you can use a filter to show members of the GOLD usergroup, select all users, use a filter to show members of SILVER and select all users again. This would in the end leave you with all members of GOLD and SILVER selected. Selected users will remain selected when you logout and return later as well.

If you want to clear a filter and also deselect any selected users then you can use the refresh  button.

### Quick Filters

Sitelok has a number of predefined quick filters that can be used to find and filter users as well as an advanced filter. The most commonly used is probably Quick Search but you can select others by click the  button.



The image shows a dropdown menu titled "Filter Type" with a downward arrow. The menu is open, showing several options. The first section is "Quick search" and includes: "All members of usergroup", "Unexpired members of usergroup", "Expired members of usergroup", "Expiring soon", "New members", and "Only selected members". The second section is "Advanced filter" and includes: "Enter SQL Query".

### Quick Search

This will show any user records containing the text entered. Its useful for finding a user quickly. You can enter partial data if you wish so for example entering dave would find all user records containing dave as part of the record in any field. Search and filter text is not case sensitive.

### All members of usergroup

This will show all users who belong to the chosen usergroup.

### Unexpired members of usergroup

This will show all users who are members of the chosen usergroup that are not expired.

### Expired members of usergroup

This will show all expired members of the chosen usergroup.

### Expiring soon

This will show all members of the chosen group whose membership will expire within the numbers of days entered. This is useful if you want to email all users who will expire within the next 30 days for example.

### New members

This will show all new user accounts created within the last chosen number of days.

### Only selected members

This will show only the users that are selected / unselected.

## Advanced filter


Click **Filter Type** and select **Advanced filter** to open the form. The advanced filter allows you to show users based on more complex requirements. For example to find all email addresses that use yahoo.com or gmail.com you would use the following search conditions.

The screenshot shows the 'Advanced filter' interface. At the top, there is a header 'Advanced filter' with a 'Filter Type' dropdown menu. Below this, there are two rows of filter conditions. The first row has a dropdown menu set to 'Email', a second dropdown menu set to 'contains', and a text input field containing 'yahoo.com'. The second row has a dropdown menu set to 'OR', a third dropdown menu set to 'Email', a fourth dropdown menu set to 'contains', and a text input field containing 'gmail.com'. To the right of the second row is a red 'X' icon. At the bottom left, there is a button labeled 'add condition' with a plus-minus icon.


The following conditions can be used to compare the search data against the data stored in the database. Equals, not equal to, contains, does not contain, less than, greater than, starts with, ends with, less than num,

greater than num, less than date and greater than date (for custom fields that store dates).

These operations are not case sensitive. When entering expiry dates ensure you enter them in the 6 digit format you selected in the configuration at installation (MMDDYY or DDMMYY).

To add a condition click  and select whether the new condition will be used with AND or OR logic. Click  to remove a condition.

### Enter SQL query

To perform more complex database queries you can enter SQL commands directly. To access this click  next to SQL Query in the filter area. We do not perform any error checking so we recommend using this feature only if you have experience due to the risk of data loss. This example query would disable all users that are members of the CLIENT usergroup.

```
UPDATE sitelok SET Enabled='No' WHERE Usergroups LIKE '%CLIENT'
```

You can disable the SQL Query filter option by adding this setting to sconfig.php.

```
$ssl_noqueryoption=true;
```

### Operations on multiple users

Sitelok enables you to perform a number actions on selected users using the buttons above the main user table on the dashboard. You can select individual users by clicking the users checkbox. The records will remain selected until you deselect them.


#### Select All users

Click the  button to select all users. If you have filtered users as just described then this will select only those records that met the conditions of the filter or search in addition to any users already selected.


#### Deselect all users

Click the  button to deselect all records. If you have filtered users then only those users will be deselected.


## Refresh

To clear any filter and also deselect all users (shows all users again) click the  button. This is useful when you start a new filter or search.


## Email selected users

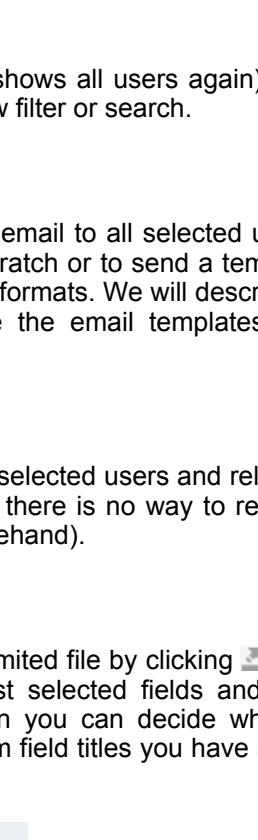
Clicking the  button will allow you to send an email to all selected users. You have the option to create the email from scratch or to send a template based email and you can use either text or html formats. We will describe in further detail later in the manual how to use the email templates and features.

## Delete selected users

Clicking the delete button  will delete all of the selected users and relevant log entries. Be very careful with this feature as there is no way to retrieve lost data (apart from using a backup taken beforehand).

## Export Selected users

You can export selected users as a comma delimited file by clicking . You can choose whether to export all fields or just selected fields and also whether a header row is included. If it is then you can decide whether custom fields will have headers using the custom field titles you have set or just custom1, custom2 etc.



**Export selected users**

**Filename**  
sitelokusers.csv

**Include header row**  
Use custom field number

**Data to export**  
Default

## Importing Users

If you wish to add several users to Sitelok then the import user feature can be useful. This can be accessed in the menu [User actions - Import users](#).

### Default file format

The default file format should be comma delimited file as follows.

```
"username","password","enabled","name","email","usergroups","custom1","custom2"
```

The enabled field should be either Yes or No. You can create the file in Microsoft Excel and save it in CSV format. The custom1 to custom50 fields are also optional and you only need to include the custom fields you are using. Example file entry:-

```
"adrianjones","wfdg45","Yes","Adrian Jones","adrian@vibralogix.com","A1","cus1","cus2"
```

If you use the default file format you do not need to define column headers.

### Custom file formats

You can also import comma delimited files with a different layout if you wish. In that case you should use the first row of the file to define the columns. Use the following header titles to tell Sitelok the data in each column.

username  
password  
name  
email  
enabled  
usergroups  
custom1  
custom2  
etc

Values should be Yes or No

Instead of [custom1](#) or [custom2](#) etc you can use the custom field titles instead.

There are a number of settings that can be set to change how new users and existing users are handled.

Import file

Select file (max 32M)

Browse...

First row is column header

Use email column as username

Never

You should check the **First row is column header** as necessary.

### New users

Records for new users should include at least an email and name field. If a username is not included select the appropriate option from the **Use email column as username** setting.

#### Add new users

Check this box to process new users from the file.

#### Random passwords

If you wish you can have Sitelok create a random password. This can be done when either no password is provided or to override the password in the file.

#### Usergroups if not set

If usergroups are not provided in the file you can set the group(s) and expiry date(s) to use.

#### Select users added

If this box is checked then all users added to Sitelok will be selected. This allows you to easily email them for example.

### New users

Add new users

#### Random passwords

Yes - ignore any provided password

#### Usergroups if not set

usergroup

days or DDMMYY



Select users added

## Existing users

### Update new users

Check this box to process existing users from the file.

### Blank data in columns

This setting tells Sitelok what to do with blank data in columns for existing users. You can either have the field set to blank or you can leave the existing data as it is already in the users account. This is useful when you want to just update certain custom fields for example.

### Select users updated

If this box is checked then all users updated will be selected. This allows you to easily email them for example.

### Existing users

Update existing users

#### Blank data in columns

Update - set field blank

Select users updated

## The Sitelok Log

Sitelok maintains a log of user actions such as login and logout etc. To access the log select **Log** form the main menu.

You can select the type of log entries to view, export or delete. Selection can be based on a date range, log entry type, a specific user or selected users.

### Date Range

Date range

**DDMMYY or leave either blank**

200816		to	
--------	---	----	---

Use the from and to date fields to select a range of log entries to view export or delete. You can enter dates manually (using DDMMYY or MMDDYY as selected in the configuration) or use the calendar controls. Leaving the from date blank will assume the first entry in the log and leaving the to date blank will assume the last entry.

### Users

If you are interested in only the details for one particular user then enter the username in the Username field. You can also choose to include only selected users or to hide selected users.

Users

**Only include username**

Leave blank for all users

**Include selected users only**

**Don't include selected users**

### Entries to include

Entries to include

Select/deselect all

Login / logout

Login problems

Password requested

Download

Download problems

Email sent

User registered

Expired access attempt

User modified details

API function call

You can select which type of log entries to display or export using the checkboxes. By default Sitelok logs the following actions by users in the log.

- Login / Logout
- Login Problems
- Password Requested
- Download
- Download problems
- Email sent
- User Registered
- Membership expired (user attempted to access)
- User modified details
- API function call

Some plugins may add other log entry types as well.

You can disable the logging of some of these actions in [Tools - Configuration - Log settings](#).

### Log output

#### Order

You can select whether to display export entries oldest first or newest first.

#### Limit Output



As the log can become huge quite quickly on busy sites you can limit the number of entries displayed or exported using this menu.

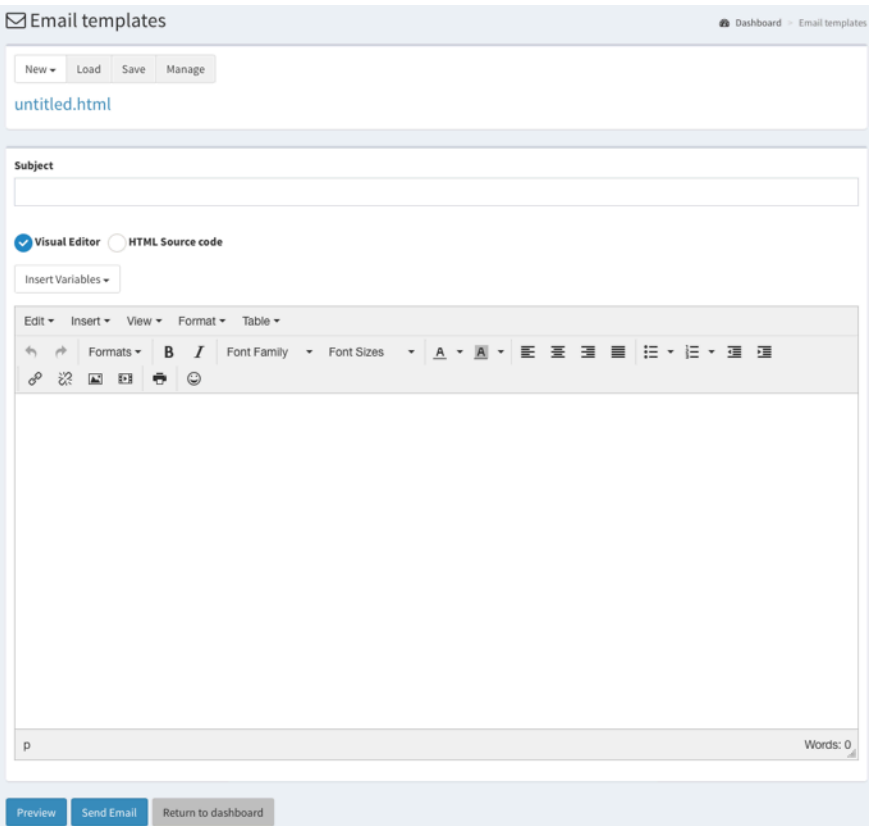
#### Time Zone

Sitelok stores all log entries using UTC (GMT) time zone which makes things easier if you move from one server to another in different time zones. So that you can display or export entries in your local time zone you can select the offset using the menu. This is the time zone used when displaying recent activity for a user too.

## Email system

The control panel allows you to email individual users or any number of selected users. This is useful to send out newsletters or other notifications to your members. Sitelok's filtering and selection features are more powerful than many dedicated mailing systems so we have lots of clients that use Sitelok just for this feature!

To send an email to a specific user click  next to the user in the main table. To send an email to selected users click the  button above the table (visible only when users are selected). You can also send an email to any email address (not just users) by clicking [Email Templates](#) in the main menu. In this case the email address will be requested when you click [Send](#) (some variables won't be of use in this case though).



There are two ways to create an email to send. The first is to enter the subject and the email body directly. The second method is to load an email template by clicking **Load** and selecting the template from the file manager.

### **Email templates**

Email templates are used throughout Sitelok when emails need to be sent to users. They contain the subject and body content and wherever user specific data needs to be inserted we use template variables. So for example we can create a template to welcome a new user and use **!!!username!!!** where we want the users username to be inserted.

Sitelok allows you to send email in either text or html format. We suggest using html format as this allows much more control over the design of the email and works with nearly all email clients nowadays. If you prefer text format then click **New - Text format**

Email templates and any associated images etc are stored in the email folder (inside the slpw folder). You can also create subfolders inside the email folder to help organize your templates and images etc. Remember that the email folder is publicly visible as referenced images will be accessed from there.

### **Loading a template**

Click Load and select the template to load from the file manager. Upon loading a template the editor will switch to text or html mode as necessary.

### **Saving a template**

Save any newly created or modified email templates by clicking the **Save** button. You will be prompted for a filename which should use the .html or .htm extension for html emails and .txt for text format emails. Any existing file with the same name will be overwritten. If you wish to save the template in a subfolder then enter the file name like this

[subfolder/template.html](#)

### **Managing templates and associated files**

The file browser can be used to manage templates and other related files. You can upload, delete and rename files as well as creating and deleting sub folders. Click the **Manage** button to open the file manager. You can

perform the same options when the file manager is open to load a template too if you wish.

### Previewing emails

We recommend using the **Preview** button to check the layout before sending the email(s). If you are sending an email to a specific user then the preview will use the users data wherever template variables are used. If you are sending to multiple users then the first selected user will be used for the preview.

### Sending emails


If you are sending an email to a specific user then it will be sent when you click the **Send** button. If you are sending an email directly from the **Email Templates** menu option you can enter any email address or email address when you click **Send**.

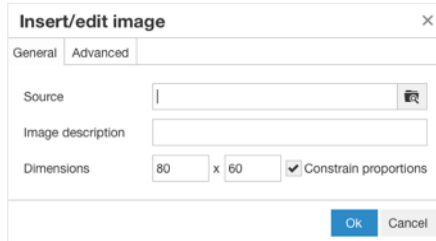
If you are sending emails to multiple selected users when you click the Send Email button a dialog will open with some sending options. There is an option to deselect users as they are emailed. This can be useful if you need to cancel the email process as you can continue later without emailing the same users again. You can also block multiple emails being sent to the same email address by selecting the **dedupe** checkbox. This is only necessary if you have users that have registered multiple times and have the same email address. When you click the **Send** button a window will open that shows the progress of the mailing. It is very important that you do not close this window or perform other operations using the control panel until the mailing has finished. How long the mailing takes depends on your server.

Some servers may have restrictions on how fast you can send multiple emails out. In **Tools - Configuration - Email Settings** you can set the delay in seconds between each email.

### HTML editor

Sitelok uses the TinyMCE editor to allow you to visually create html emails. You can click the **HTML Source code** radio button to view and edit the html code directly as well. TinyMCE is easy to use and works in a similar way to other editors you have probably worked with before.

The file manager has been integrated with the editor so that you can upload and select images or other media for use in the email. You can access the file manager when adding media by clicking the  button in the media dialog.



## Email template variables

So that you can personalize each email that is sent out Sitelok has a set of template variables that you can use. These variables will be replaced by the users actual data when the email is sent. These variables work in templates and if you directly type in an email without saving it. For example if you type the variable **!!!username!!!** it will be replaced by the users actual username when the email is sent. As well as simply being placeholders for user data variables can also generate secure download links, login links, approval links etc.

### Inserting variables into an email

You can either type a variable directly into the email (visual editor, source editor or the plain text editor) or you can select it from the drop down list displayed by clicking the **Insert Variables** button. This will insert the variable at the current cursor location.

Some variables that generate links (such as **!!!link(test.zip,1440)!!!**) can either be added directly to the email in which case the full link is seen or added to a hyperlink which is usually better. To do this enter the hyperlink text, select it and then insert the variable.

Some variables may have a number of parameters in which case you can either type these directly or enter them in the dialog box that appears when you use the **Insert Variables** drop down.

### General template variables

**!!!username!!!**                      Inserts the username

<b>!!!password!!!</b>	Inserts the password (not for hashed passwords)
<b>!!!passwordclue!!!</b>	Inserts the last three characters of the password (not used if you enable hashed passwords)
<b>!!!passwordhash!!!</b>	Inserts a secure hash of the password
<b>!!!passwordifknown!!!</b>	If you enable hashed passwords then this variable will display the plain text password if known (after registration for example) or if already hashed will display *****. This text can be replaced by setting  <code>define("MSG_PASSWORDHASHED", "*****");</code>  in <code>slconfig.php</code> as needed.
<b>!!!name!!!</b>	Inserts the name
<b>!!!firstname!!!</b>	Inserts the first name
<b>!!!lastname!!!</b>	Inserts the last name
<b>!!!email!!!</b>	Inserts the email address
<b>!!!groups!!!</b>	Inserts the data from the <code>usergroups</code> field
<b>!!!sitename!!!</b>	The site name as entered in the configuration
<b>!!!siteemail!!!</b>	The site email address as entered in configuration
<b>!!!custom1!!!</b>	Inserts the data from the <code>custom1</code> field. You can display values from <code>custom1</code> to <code>custom50</code> .
<b>!!!date!!!</b>	Inserts the date as <code>dd/mm/yy</code> or <code>mm/dd/yy</code> based on the site date format setting.
<b>!!!datedmy!!!</b>	Inserts the date as <code>dd/mm/yy</code>
<b>!!!datemdy!!!</b>	Inserts the date as <code>mm/dd/yy</code>
<b>!!!createdmy!!!</b>	Inserts the users account creation date
<b>!!!createdmdy!!!</b>	Inserts the users account creation date
<b>!!!userid!!!</b>	Inserts the users unique numeric id

These variables will be replaced with the data from each users record as the email is sent. As the `groups` field above can look a bit messy you can use the following feature to add the membership group details to the email. This works by defining a section of the email body that will be repeated for each usergroup the user is a member of. This enables you for example to list each group and the expiry time that the user is a member of. These tags define the start and end of this repeated section.

<b>!!!eachgroupstart!!!</b>	This defines the beginning of the section to repeat.
<b>!!!eachgroupend!!!</b>	This defines the end of the section to repeat.

Between these two tags you can use the following variables.

!!!groupname!!!	The group name such as SUPPORT or A1 etc.
!!!groupdesc!!!	The full group description.
!!!groupexpiry!!!	The expiry date shown as MM/DD/YY or DD/MMYY depending on the setting in the configuration.
!!!groupexpirydmy!!!	The expiry date shown as DD/MMYY.
!!!groupexpirymdy!!!	The expiry date shown as MM/DD/YY
!!!groupremove!!!	This inserts a link that when clicked will remove the user from that group. Normally you would use this as a hyperlink such as <code>&lt;a href="!!!groupremove!!!"&gt;Remove&lt;/a&gt;</code>

You can define the group description displayed when using the variable **!!!groupdesc!!!** in the **Manage Groups**.

### Auto login links

Sometimes you may wish to include a link in an email which points to a secured page where you want to login the user automatically (no login box)). To do this generate the link like this

[http://www.yoursite.com/page.php?username=!!!username!!!&password=!!!passwordhash\(1440\)!!!](http://www.yoursite.com/page.php?username=!!!username!!!&password=!!!passwordhash(1440)!!!)

where page.php is a secured members page. Notice that we are sending the password hash rather than the actual password to increase security. These links can bypass Captcha if it is enabled.

The 1440 is a time limit on the link from when it was sent in minutes (1440 is 24 hours). You can set this to 0 for no time limit although its usually best to have some limit.

The easiest way to insert this variable is to select the hyperlink text for the link and then click **Auto Login link...** from the **Insert Variables** drop down. You can then enter the parameters in the dialog box.

**Remember** that although auto login links don't reveal the users password they could still be intercepted and used until they expire.

### Secure download links

You can include secure download links in emails that you send allowing users access to files that you have uploaded to the folder created during installation with a name like sfiles\_XXXX or any additional file locations you

have defined in **Tools - Configuration - Download Locations**. These links do not show the location of the file on your server and can be time limited. The links generated are unique for each email sent and username. Here is an example that assumes you have uploaded test.zip to the correct folder (see the example email template download.htm for reference).

**!!!link(test.zip,1440)!!!** This generates a download link for test.zip that is valid for 1440 minutes (24 hours). You can set 0 for no expiry. It is also possible to set a fixed expiry date by entering the expiry as 12 digits in the format **yyymmddhhmm**. So for example to have an expiry time of August 31st 2010 at 23:59 you would set the expiry time as 201008312359.

**!!!size(test.zip)!!!** Inserts the file size of test.zip.

If you have enabled logging of downloads you will see which recipients have downloaded files.

If you have created additional download locations you can specify the location in the link by adding **:** and the location name after the filename. For example if you create a location called S3 you would specify this as

**!!!link(test.zip:S3,1440)!!!**

The easiest way to insert this variable is to select the hyperlink text for the link and then click **Download link...** from the **Insert Variables** drop down. You can then enter the parameters in the dialog box.

### **Links to enable, disable and delete user accounts**

As these variables create links the easiest way to insert them is to select the hyperlink text for the link and then click the required variable from the **Insert Variables** drop down. You can then enter the parameters in the dialog box.

**!!!approve( )!!!**

This will insert a link that when clicked will enable the users account. This could be used within an admin email sent after a user registers so that you can manually approve the user (the registration form would create the account initially disabled). You could instead have this in the users registration email so that they can confirm their email is correct and enable

their account. When the link is clicked it will enable the account and optionally send email templates to the user and admin. It will then redirect to the optional redirect URL. See the example called `pendingadmin.htm` for the use of `!!!approve()!!!`. Also see the example registration form that requires approval called `registerapprove.php` in the `members` folder. Remember if you use an `approve` link in an admin email never to forward this to a member by accident!

```
!!!approve(expiry,clienttemplate,admintemplate,page)!!!
```

<code>expiry</code>	This is the validity of the link in minutes.
<code>clienttemplate</code>	The email template to send to the user when clicked. Leave blank for no email
<code>admintemplate</code>	The email template to send to the admin when clicked. Leave blank for no email.
<code>page</code>	The full URL of the page to go to after the link is clicked. If you leave this blank a simple message is displayed.

```
<a href="!!!approve(1440,approve.htm,approveadmin.htm,http://www.yoursite.com/index.htm)!!!">Click to confirm</a>
```

This would create a link valid for 1440 minutes (1 day), send `approve.htm` to the user and `approveadmin.htm` to admin and then redirect to `http://www.yoursite.com/index.htm`.

### !!!disable!!!

This will insert a link which when clicked will disable the account and optionally send email templates to the user and admin. It will then redirect to the optional redirect URL.

```
!!!disable(expiry,clienttemplate,admintemplate,page)!!!
```

<code>expiry</code>	This is the validity of the link in minutes.
<code>clienttemplate</code>	The email template to send to the user when clicked. Leave blank for no email
<code>admintemplate</code>	The email template to send to the admin when clicked. Leave blank for no email.
<code>page</code>	The full URL of the page to go to after the link is clicked.

### example usage

```
<a href="!!!disable(1440,disable.htm,disableadmin.htm,http://www.yoursite.com/index.htm)!!!">Click to disable account</a>
```

This would create a link valid for 1440 minutes (1 day), send `disable.htm` to the user and `disableadmin.htm` to admin and then redirect to `http://www.yoursite.com/index.htm`.

### !!!delete( )!!!

This will insert a link which when clicked will delete the account and optionally send email templates to the user and admin. It will then redirect to the optional redirect URL.

```
!!!delete(expiry,clienttemplate,admintemplate,page)!!!
```

<code>expiry</code>	This is the validity of the link in minutes.
<code>clienttemplate</code>	The email template to send to the user when clicked. Leave blank for no email
<code>admintemplate</code>	The email template to send to the admin when clicked. Leave blank for no email.
<code>page</code>	The full URL of the page to go to after the link is clicked.

### example usage

```
<a href="!!!delete(1440,delete.htm,deleteadmin.htm,http://www.yoursite.com/index.htm)!!!">Click to permanently delete account</a>
```

This would create a link valid for 1440 minutes (1 day), send `delete.htm` to the user and `deleteadmin.htm` to admin and then redirect to <http://www.yoursite.com/index.htm>.

## Links to adjust usergroups

This section explains the email template links that can be used to add or remove users to usergroups.

As these variables create links the easiest way to insert them is to select the hyperlink text for the link and then click the required variable from the **Insert Variables** drop down. You can then enter the parameters in the dialog box.

### !!!addgroup( )!!!

This will generate a link in the email which when clicked will add the user to the specified usergroup.

!!!

```
addgroup (expiry,group,grouppexpiry,clienttemplate,admin  
emplate,page) !!!
```

<b>expiry</b>	This is the validity of the link in minutes.
<b>group</b>	The group to add the the user to
<b>grouppexpiry</b>	The expiry for the group. This can be in days (e.g. 365) or a fixed six digit date. Here are some examples <b>365</b> This would set the new group with an expiry 365 days from now <b>010809</b> This would set the expiry to 1st August 2009 assuming DDMMYY format
<b>clienttemplate</b>	The email template to send to the user when clicked. Leave blank for no email
<b>adminemplate</b>	The email template to send to the admin when clicked. Leave blank for no email.
<b>page</b>	The full URL of the page to go to after the link is clicked.

example usage

```
<a href="!!!addgroup(1440,GOLD,365,user.htm,admin.htm,http://  
www.yoursite.com)!!!">Join GOLD group</a>
```

### !!!removegroup( )!!!

This will generate a link in the email which when clicked will remove the user from the specified usergroup.

```
!!!removegroup (expiry,group,clienttemplate,adminemplate,page) !!!
```

<b>expiry</b>	This is the validity of the link in minutes.
<b>group</b>	The group to remove the user from
<b>clienttemplate</b>	The email template to send to the user when clicked. Leave blank for no email
<b>adminemplate</b>	The email template to send to the admin when clicked. Leave blank for no email.
<b>page</b>	The full URL of the page to go to after the link is clicked.

example usage.

```
<a href="!!!removegroup(1440,GOLD,user.htm,admin.htm,http://www.yoursite.com)!!!">Leave GOLD group</a>
```

### !!!replacegroup( )!!!

This will generate a link in the email which when clicked will replace one group with another for the user.

```
!!!replacegroup
expiry,group,newgroup,groupepiry,clienttemplate,admintemplate,page)!
!!
```

<code>expiry</code>	This is the validity of the link in minutes.
<code>group</code>	The group to be replaced
<code>newgroup</code>	The new group to replace the old one
<code>groupepiry</code>	The expiry for the new group. Leave blank to retain current expiry date. Here are some examples
<code>365</code>	This would set the new group with an expiry 365 days from now
<code>010809</code>	This would set the expiry to 1st August 2009 assuming DDMMYY format
<code>+30</code>	This would extend the current expiry by 30 days
	Leave blank This will leave the current expiry date
<code>clienttemplate</code>	The email template to send to the user when clicked. Leave blank for no email
<code>admintemplate</code>	The email template to send to the admin when clicked. Leave blank for no email.
<code>page</code>	The full URL of the page to go to after the link is clicked.

### example usage

```
<a href="!!!replacegroup(1440,SILVER,GOLD,365,user,htm,admin.htm,http://www.yoursite.com)!!!">Upgrade from SILVER to GOLD group</a>
```

### !!!extendgroup( )!!!

This will generate a link in the email which when clicked will extend the expiry date.

```
!!!
extendgroup(expiry,group,groupepiry,expirytype,clienttemplate,admintemplate,page)!!!
```

<code>expiry</code>	This is the validity of the link in minutes.
<code>group</code>	The group to be extended
<code>groupexpiry</code>	The expiry for group. Here are some examples
<code>365</code>	This would set the new group with an expiry 365 days from now
<code>010809</code>	This would set the expiry to 1st August 2009 assuming DDMMYY format
<code>+30</code>	This would extend the current expiry by 30 days
<code>expirytype</code>	This setting can be used to determine whether to extend the expiry. Here are some examples
<code>0</code>	Setting 0 will always adjust the expiry date
<code>+30</code>	Using a + in front of the number will only adjust the expiry date if the existing expiry was 30 (in this case) or more days ago.
<code>-30</code>	Using a - in front of the number will only adjust the expiry if there are at least 30 (in this case) days until the current expiry.
<code>clienttemplate</code>	The email template to send to the user when clicked. Leave blank for no email
<code>admintemplate</code>	The email template to send to the admin when clicked. Leave blank for no email.
<code>page</code>	The full URL of the page to go to after the link is clicked.
<code>example usage</code>	

```
<a href="!!!extendgroup(1440,GOLD,365,+30,user.htm,admin.htm,http://ww.yoursite.com)!!!">Extend GOLD membership</a>
```

### Link to generate a new password

**!!!newpassword!!!**

This will display a new password which will be activated when the user clicks the link displayed using **!!!activatepassword!!!**. This can be useful in a forgotten password email template where instead of displaying the existing password a new random one is used. See the example email template called newpassword.htm.

## Confirmation of email change

If you enable 'Email change verification' you can use these variables in the related email templates.

!!!verifyemail!!!

This inserts a link that when clicked will update the email address. This should normally be used in a hyperlink like this

```
<a href="!!!verifyemail!!!">click to confirm</a>
```

!!!requestedemail!!!

The requested password.

See the example template called verifyemail.htm. For further information see the chapter called 'User Profile Page'.

## Allowing a user to view the email in a their browser

You can add a link to your email template which will allow the user to view the content in a browser instead of the email client. To do this use

!!!viewemail(0,template.html)!!!

This will create a link to display template.html (replace with your template filename). The 0 means that there is no time limit on the link but you can set this to a time in minutes (since sending) if you wish. Normally you would use this variable as part of a hyperlink like this

```
<a href="!!!viewemail(0,template.html)!!!">Click to view in your browser</a>
```

## Sending the email(s)

If you are sending email to one user or an entered email address it will be sent when you click the Send Email button.

If you are sending emails to multiple selected users when you click the Send Email button a small window will open that shows the progress of the mailing. It is very important that you do not close this window or perform other operations using the control panel until the mailing has finished. How long the mailing takes depends on your server.

There is an option to deselect users as they are emailed. This can be useful if you need to cancel the email process as you can continue later without emailing again the same users.

Some servers may have restrictions on how fast you can send multiple emails out. In the configuration page you can set the delay in seconds between each email.

You can also block multiple emails being sent to the same email address by selecting the **dedupe** checkbox on the email form. This is only necessary if you have users that have registered multiple times and have the same email address.

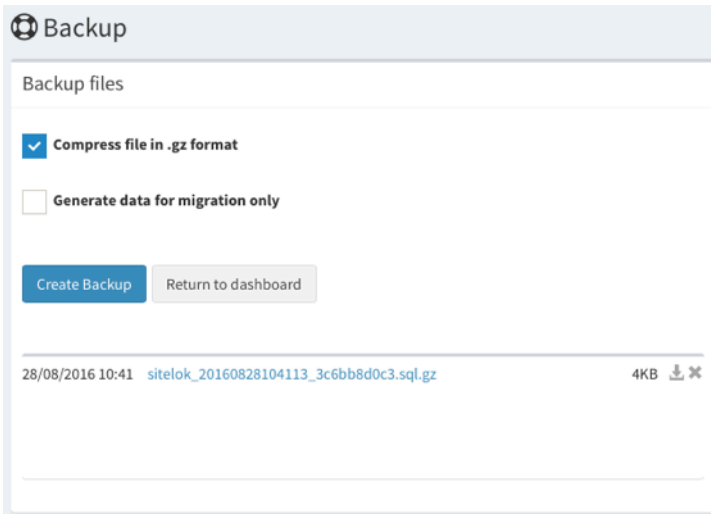
There is an option to deselect users as they are emailed. This can be useful if you need to cancel the email process as you can continue later without emailing again the same users.

### **Creating email templates using an external editor**

If you wish to create email templates using an external editor you can do so. You can either upload them to the email folder using your usual FTP client or using the upload button on the file browser used on Sitelok email page. Alternatively you can paste them into the Sitelok email editor and save them. Use the .txt extension for text based emails and .htm for html based templates. For text format the first line of the email is considered to be the email subject and the following lines the email body. For html format the html title is considered the email subject (the text between the <title> and </title> tags). We have included several example emails in the email folder which you can use or modify as required.

## Backing up

Backing up is always an important task and although most hosting control panels provide a way to backup MySQL tables we decided to add this directly to Sitelok to make this a simple process. Click **Tools - Backup** from the main menu to open the backup page.



Clicking the Backup button will create a new backup file that will be stored in a folder create during installation. This will be named slbackups\_XXXX (the part after \_ will be random). The backup system can handle large databases and will store the data required to recreate all users and settings used by Sitelok. The file format used is .sql so the file can be imported into any standard mysql admin system such as phpmyadmin. There is an option to compress the file after creation using gzip which will dramatically reduce the file size. Once the file has been created and saved you can download it by clicking the file link. We recommend deleting the files from the server after downloading them for security.

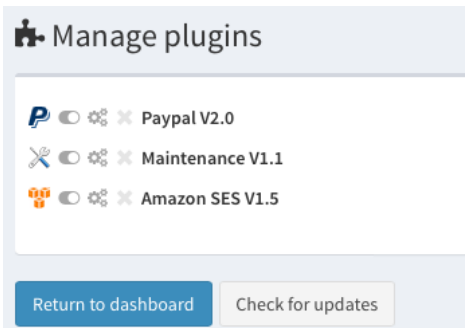
If you wish to create a backup file so that you can migrate your Sitelok installation to a new server you can check the 'Generate data for migration only' box. This will generate a backup that is almost the same as normal except that it exclude things like server paths etc that are unique to the new server.


**Important** You should always use a backup file with the same version of Sitelok. If you wish to migrate then make sure the new installation has the same version that created the backup file.


## Plugins


Sitelok has a plugin system enabling new features to be added. We have many free and paid plugins released and more on the way. To install plugins you should follow the instructions in the manual that comes with the plugin package.

You can enable/disable, configure, and uninstall plugins from the Manage plugins page accessed from the menu [Plugins - Manage plugins](#).



Click  next to a plugin to disable it (or re-enable).

Click  next to a plugin to go to the plugin configuration page.

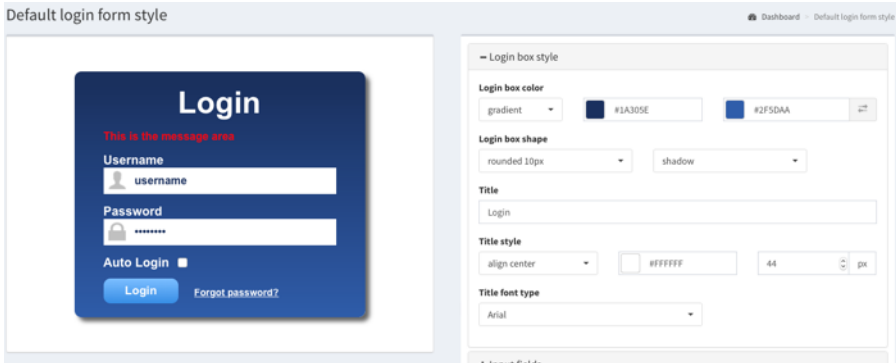
Click  next to a plugin to uninstall it. You must disable the plugin first.

You can check for updates for plugins by clicking the [Check for updates](#) button.

Some plugins may also add extra menu options, user options or selected user options.

## Styling the default login template

You can style the default login form to better match your site by going to [Forms - Default login form style](#).



If you need more creative control then you can also replace the login template completely with your own as explained later.

### Registration form code

Go to [Forms - Registration forms](#) to design a registration form and generate the code to copy and paste in your page. See the chapter 'User Registration Page' for more details.

### Update profile form code

Go to [Forms - Update profile forms](#) to design an update profile form and generate the code to copy and paste in your page. See the chapter 'User Profile Page' for more details.

### Login form code

Go to [Forms - Login forms](#) to design a login form and generate the code to copy and paste into a custom login template or your own pages. See the chapters 'Login Templates' and 'Login Pages' for more details.

### Contact form code

Go to [Forms - Contact forms](#) to design a login form and generate the code to copy and paste into a custom login template or your own pages. See the chapters 'Contact forms' for more details.

## **SUBADMIN access**

Sitelok has two types of sub admin user.

### **Standard SUBADMIN**

If you create a user and set their usergroup to SUBADMIN then they will have access to the Sitelok control panel but not be allowed to perform the following

- Edit or see the password of an ADMIN user (including in email preview)
- Add an ADMIN user
- Access plugin settings (user specific plugin settings can be accessed)
- Access the configuration page
- Import users in the ADMIN group
- Delete selected which is quite dangerous (they can delete individual users though)
- Export users or create a backup (as they could then download and see the admin password)
- Access the SQL query function

The sub admin user will have access to all users in the control panel.

### **Usergroup specific SUBADMIN**

As well as the standard SUBADMIN user you can also create users who can only see and edit users of specific usergroups in the control panel. To enable this feature you first need to add this setting to slconfig.php.

```
$sl_subadmingroups=true;
```

Then create a user in the SUBADMIN usergroup and one or more other usergroups that they will have access to.

The user will have access to the Sitelok control panel but not be allowed to perform the following

- Edit or see the password of an ADMIN user (including in email preview)
- Add an ADMIN user or users in usergroups they have no access to
- Access plugin settings (user specific plugin settings can be accessed)
- Access the configuration page
- Import users in the ADMIN group or usergroups they have no access to
- Export users or create a backup (as they could then download and see the admin password)

- Access the SQL query function
- Select users or perform actions on selected users. This is because selecting users is global and would affect the main ADMIN users access.

### Translating the control panel

You can change the wording or translate any of the text used in the control panel. By default all of the English text is stored in the file

`adminlanguage-en.php`

which you can adjust as needed. For example the text 'User actions' is defined in the line

```
define("ADMINMENU_USERACTIONS", "User actions");
```

Just change the text (shown in bold).

If you are translating to a new language you may prefer to rename the `-en` part of the file name to `-de` for example. You can then instruct Sitelok to load this language file instead by adding this setting to `slconfig.php` (anywhere is ok).

```
$adminlanguage="de";
```

Set this to `en` or delete the setting to revert to the default English file.

When you update Sitelok you can normally keep your existing language file. Any new or changed text is shown in the new file so you can add that in as needed

You can also white label Sitelok if you wish which is ideal if you are a web designer and use Sitelok on your clients sites. You are free to change all of the text including the visible copyright message in the control panel although you must leave any copyright messages within the code in place. You can also change the logo images shown in the top left of the control panel.

`/slpw/adminlte/img/adminlogo.png`

Main logo

`/slpw/adminlte/img/adminlogomini.png`

Mini logo used when navigation is minimized

The theme color can be changed by adding this setting to slconfig.php (anywhere is ok)

```
$adminskin="blue";
```

You can set this to blue (this is the default), blue-light, black, black-light, green, green-light, purple, purple-light, red, red-light, yellow or yellow-light.

If you wish to disable the SQL Query filter option you can add this setting to slconfig.php

```
$sl_noqueryoption=true;
```

## Chapter 5 Securing Pages

Protecting pages with Sitelok is easy. Just a small code snippet added to the top of the page will secure the page and provide lots of features.

Sitelok can only protect PHP pages so if your site uses pages with the extension .htm or .html you should rename your pages to have the .php extension. This will have no affect on the appearance of your site but allows Sitelok to control the pages. Most servers can be set to automatically handle html pages as PHP if renaming your pages is a problem. This usually just requires adding a line to the .htaccess file on your site. Contact your hosting company who should be able to help with that.

### Allowing access to certain usergroups

In most cases you will probably wish to provide access to pages only to users of certain usergroup(s). To do this simply add this code snippet to the very top of the page to secure.

```
<?php
$groupswithaccess="GROUPNAME";
require_once("slpw/sitelokpw.php");
?>
```

**GROUPNAME** should be replaced by the name of the usergroup you wish to have access to the page. If you wish all groups to have access then use the name **ALL**.

You can allow access to multiple groups by listing them separated by commas such as `$groupswithaccess="GOLD,SILVER,BRONZE";`

Like any links on your website The path to sitelokpw.php should be the correct relative path to the sitelokpw.php in the slpw folder. Here are some examples

<code>slpw/sitelokpw.php</code>	Is the path for a page in the root of your site
<code>../slpw/sitelokpw.php</code>	Is the path for a page in a folder one level down.
<code>../../slpw/sitelokpw.php</code>	Is the path for a page in a folder two levels down.
<code>../../../slpw/sitelokpw.php</code>	Is the path for a page in a folder three levels down.

You can see two examples of secured pages in the example members area. These are members.php and members2.php

## Allowing access to certain users only

Normally you would protect pages for certain groups which would allow access for any user who belongs to at least one of those groups. There may however be applications where you want to secure access to pages on a user basis. To do this add the following code snippet to the very top of the page.

```
<?php
$userswithaccess="USERNAME";
$groupswithaccess="GROUPNAME";
require_once("/slpw/sitelokpw.php");
?>
```

**USERNAME** should be replaced by the username of the user you wish to have access to the page. You must also set the **GROUPNAME** in which you are allowing this user to have access. This is important for expiry control for example because users can be members of several groups.

You can allow access to multiple users by listing them separated by commas such as `$userswithaccess="username1,username2";`

## Adding a logout link

A users session will end when they exit the browser. However its a good idea to add a logout button that enables the user to end a session whenever they wish which is more secure.

To add a logout you just need to create a link or button that points to

</slpw/sitelokpw.php?sitelokaction=logout>

For example:-

```
<a href="/slpw/sitelokpw.php?sitelokaction=logout">Logout</a>
```

When a user clicks this link they will be logged out (their session cancelled) and then get redirected to the page you set for Logout Page in the configuration.

If you wish to override the default logout page you can pass that in like this

```
<a href="/slpw/sitelokpw.php?sitelokaction=logout&page=http://www.yoursite.com">Logout</a>
```

### Securing download links

Unlike .htaccess protection Sitelok is only securing the pages that you have added Sitelok to. This makes for a flexible system but has the disadvantage that other download files or graphics that are embedded in the page or linked to are not secured and could be accessed without logging in if the URL is known. To solve this problem Sitelok uses the Linklok system to secure links to such files. These links securely hide the true location of the file and only work for users that are logged in and authorized. The link also cannot be tampered with (filename changed for example) as this will be detected by the hash system embedded in the link.

Files that need to be secured in this way should be uploaded to the folder pointed to by the **Full file path to download folder** setting in **Tools - Configuration - Download paths**.

#### Download Locations

##### Full file path to download folder

```
/home/public_html/slfiles_3r2t6dg32ut
```

During installation a random folder is created in the root of your site but you can change this if required. This folder can also have as many subfolders as you wish which may help organise files for different usergroups etc..

To make a link secure you need to replace the url with a call to the PHP function `siteloklink()` as follows (see download.php in the members folder for reference).

#### Original link

```
<a href="../files/testfile.zip">">Download here</a>
```

#### Sitelok secured link

```
<a href="<?php siteloklink('testfile.zip',1); ?>">Download here</a>
```

The first parameter is the filename (testfile.zip) which should be stored in the folder we mentioned above.

The second parameter tells Sitelok whether this file is to be downloaded or displayed inline. Setting it to **1** means download and is normally used with HREF. Setting it to **0** includes the file inline and is normally used with SRC.

Although most of the time using one folder and subfolders will be sufficient it is also possible to define multiple folders to store your files. This is useful if perhaps you have some files located on remote servers. To do this just add the extra location in the configuration page and provide a label for each one. For example

### Additional download locations

files	/home/public_html/files879452gdw/
images	/home/public_html/images/r2ewfhjtyu

You can create as many locations as you need naming them as required. In this case we named the locations 'files' and 'images'. To specify which location to use in the Sitelok link just add a colon and the location name after the filename. For example to download testfile.zip in the files location you would use the following.

```
<a href="<?php siteloklink('testfile.zip:files',1)?>">Download here</a>
```

### Storing download files on remote servers

As well as providing access to downloads on your server Sitelok also supports downloading files from remote servers although some hosts may not support it. To do this you still use the siteloklink() function as described but use a full URL for the folder definition in the configuration. For example

<http://www.remoteserver.com/folder/>

The last / character is required. Please note that accessing files remotely like this will use bandwidth on both the remote server and your main server where Sitelok is installed. If you are accessing large files remotely you may like to try Sitelok's Amazon S3 support instead.

### Storing download files on Amazon S3

As well as providing access to downloads on your server Sitelok also supports downloading files from Amazon S3. This works the same using the `siteloklink()` function in your pages but instead of using a path for the file location in the configuration you will use a special S3 path like this.

`s3|accesskeyid|secretaccesskey|bucketname|`

Notice this path uses the pipe character `|` to split the various parts.

<code>s3</code>	This tells Linklok that this is an S3 path.
<code>accesskeyid</code>	This is the access key id provided by S3 when you open an account
<code>secretaccesskey</code>	This is the secret access key provided by S3 when you open an account. You must not reveal this key.
<code>bucketname</code>	This is the unique bucket name where you have stored the files. <b>Bucket names must be all lowercase.</b>

### S3 V4 Signatures

By default Sitelok uses S3's V2 signatures to generate signed links. Some new bucket regions such as Frankfurt only support the newer V4 signature. These are supported by Sitelok as well by using a slightly different path which includes the region name. Here is a V4 example path

`s3|region|accesskeyid|secretaccesskey|bucketname|`

The region part should be the standard S3 region identifier such as `us-east-1`, `us-west-1`, `eu-west-1`, `eu-central-1` etc.

US Standard (N. Virginia)	<code>us-east-1</code>
US West (Oregon)	<code>us-west-2</code>
US West (N. California)	<code>us-west-1</code>
EU (Ireland)	<code>eu-west-1</code>
EU (Frankfurt)	<code>eu-central-1</code>
Asia Pacific (Singapore)	<code>ap-southeast-1</code>
Asia Pacific (Sydney)	<code>ap-southeast-2</code>
Asia Pacific (Tokyo)	<code>ap-northeast-1</code>
South America (Sao Paulo)	<code>sa-east-1</code>

You can find the full list at

[http://docs.aws.amazon.com/general/latest/gr/rande.html#s3\\_region](http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region)

For more information about S3 or to open an account please visit

<http://aws.amazon.com/s3/>

## Chapter 6 Displaying user data in pages

Within members pages you can display and make use of several predefined Sitelok variables. These are PHP variables and can therefore be inserted into your pages simply by using the echo / print function or used programmatically in your own scripts. If you use them in HTML you can style them to match your site. As PHP is processed before the page is sent to the browser you can also embed these variables within Javascript as well. You can also combine these variables with the Sitelok API functions described in a later chapter allowing very complex web applications to be developed. Here is a list of the variables available.

### **\$slusername**

Contains the current users username. To display the contents use this code snippet:-

```
<?php echo $slusername; ?>
```

### **\$slpassword**

Contains the current users password. remember that there are security implications when displaying a users password. To display the contents use this code snippet:-

```
<?php echo $slpassword; ?>
```

### **\$slpasswordclue**

Contains the last three characters of the current users password. Remember that there are security implications when displaying a users password. To display the contents use this code snippet:-

```
<?php echo $slpasswordclue; ?>
```

### **\$slname**

Contains the name of the current user. To display the contents use this code snippet:-

```
<?php echo $slname; ?>
```

### **\$slfirstname**

Contains the users first name (characters up to first space). To display the contents use this code snippet:-

```
<?php echo $slfirstname; ?>
```

## **\$sllastname**

Contains the users last name (characters after the last space). To display the contents use this code snippet:-

```
<?php echo $sllastname; ?>
```

## **\$slemail**

Contains the email of the current user. To display the contents use this code snippet:-

```
<?php echo $slemail; ?>
```

## **\$slcustom**

\$slcustom1 contains the data from the custom1 field for the current user. You can use \$slcustom1 to \$slcustom50. Please note that even if you re-title a custom field you still use these same variable names in Sitelok.

```
<?php echo $slcustom1; ?>
```

## **\$sluserid**

Contains the users unique numeric id. To display the contents use this code snippet:-

```
<?php echo $sluserid; ?>
```

## **\$slstarttime**

Contains the time the current user logged in for this session stored in the PHP time format. To display this in your page in a meaningful way use one of the following code snippets:-

```
<?php echo date("H:i", $slstarttime); ?>           HH:MM format  
<?php echo date("m/d/y H:i", $slstarttime); ?>    MM/DD/YY HH:MM format  
<?php echo date("d/m/y H:i", $slstarttime); ?>    DD/MM/YY HH:MM format
```

## **\$slaccesstime**

Contains the time the current user accessed a page during this session stored in the PHP time format. To display this in your page in a meaningful way use one of the following code snippets:-

```
<?php echo date("H:i", $slaccesstime); ?>           HH:MM format  
<?php echo date("m/d/y H:i", $slaccesstime); ?>    MM/DD/YY HH:MM format  
<?php echo date("d/m/y H:i", $slaccesstime); ?>    DD/MM/YY HH:MM format
```

## **\$slcreated**

Contains the date the current user joined the Sitelok database stored in the PHP time format. To display this in your page in a meaningful way use one of the following code snippets:-

`<?php echo date("m/d/y",$slcreated); ?>` MM/DD/YY format  
`<?php echo date("d/m/y",$slcreated); ?>` DD/MM/YY format

## **\$slusergroups**

Contains the usergroups and expiry times (if defined) for the the current user. This variable contains the data exactly as it is stored in the usergroups database field (e.g. GOLD:120605|SILVER:080205). In practice it is probably easier to use the array variables `$slgroupname`, `$slgroupdesc` and `$slgroupexpiry` described below.

## **\$slgroupname, \$slgroupdesc and \$slgroupexpiry**

These array variables contain the current users group membership details in a format that is easier to use than `$slusergroups`. These are array variables so for example `$slgroupname[0]` contains the first group name the member belongs to and `$slgroupname[1]` the second etc.

`$slgroupname[ ]` The group name the user belongs to  
`$slgroupdesc[ ]` The description of the group  
`$slgroupexpiry[ ]` The expiry date for the group in MM/DD/YY or DD/MM/YY format depending on the date setting in the configuration.  
`$slgroupexpiryts[ ]` The expiry date for the group as a PHP GMT timestamp

For example to display each of the groups a user belongs to together with the expiry time you could use the following code:-

```
<table>
<?php for ($k=0;$k<count($slgroupname);$k++) { ?>
<tr>
<td><?php echo $slgroupdesc[$k]; ?> </td>
<td><?php echo $slgroupexpiry[$k]; ?> </td>
</tr>
<?php } ?>
</table>
```

The two lines in green start and end a loop for each group the user is a member of.

## **\$slgroupexpirybyname['GROUP']**

Allows you to retrieve the expiry date for a certain usergroup in MM/DD/YY or DD/MM/YY format depending on the date setting in the configuration.

for example

```
<?php echo $slgroupexpirybyname['GOLD']; ?>
```

### **\$slgroupexpirybyname['GROUP']**

Allows you to retrieve the expiry date for a certain usergroup as a PHP GMT timestamp

for example

```
<?php echo $slgroupexpirytsbyname['GOLD']; ?>
```

### **\$slpublicaccess**

Is set to true when viewing a page with PUBLIC access if there is no logged in user or the user only has public access permission. See the section later about providing public access to part of secured pages.

### **\$sljustloggedin**

Is set to true if this is the first page view after logging in. This could be useful to display a popup window to welcome a user or display some news for example. Here is a simple example that pops up a news page upon login.

```
<?php
if ($sljustloggedin==true)
{
?>
<script language="javascript" type="text/javascript">
    window.open('http://www.yoursite.com/news.htm','news','');
</script>
<?php
}
?>
```

### **Force database read on page access**

Normally once a session has begun the users details are stored in session variables for efficiency because it saves making unnecessary calls to the mysql database. However if you were to update some of the user data (for example a custom field value) from the admin panel or via the Sitelok API then this value wouldn't be updated for display on the members pages until the user starts a new session. In most cases this is not important but you

can force Sitelok to either always read this data from the database or to do this on selected pages.

You can force Sitelok to read the user data from the database on each page access by enabling the setting Force database read on page access in the configuration page of Sitelok admin.

To simply enable this on a page by page basis just add this setting to the page before the call to sitelokpw.php.

```
$dbupdate=true;
```

## Chapter 7 User Registration Page

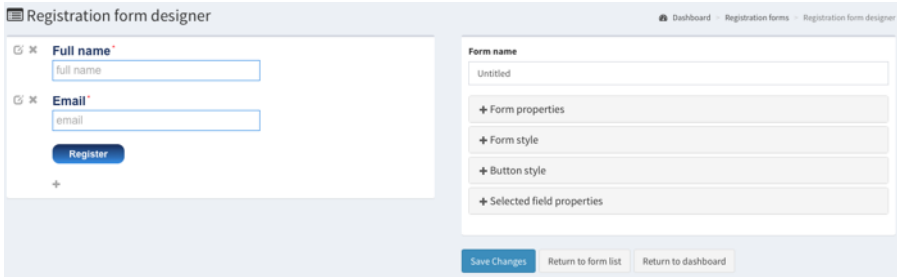
There are several ways to add users to the Sitelok system but for many sites you will want to allow them to join themselves. This is done via a registration form. Registration forms collect information about the user and then create a Sitelok account for them. You can collect as little or as much data as you want from the user and you can decide whether to provide immediate access or wait to approve each application. You can also set the period for which their membership will last (expiry time),

Before starting on your registration form give some thought to what information you need to collect. Remember asking a user for too much information may put users off so keep things as simple as possible. You can always get more information from them later on via the modify profile page. You should also consider what you wish to use for a users username and how their password is issued. Often using the users email address as the username is the easiest as this is unique and easy to remember. You can have Sitelok create a random password and email this to the user or the user can choose their own password. There are also options to have the admin approve users before they have access or have users confirm their email address.

There are two ways you can make registration forms. The easiest is to use the Registration form code feature in the control panel. This will generate the code needed so that you can copy and paste it in your page. If you are more advanced user you can also use your own html forms. Both methods are described in detail in this chapter.

## Creating a registration form using the Form Designer

To open the form designer select **Forms - Registration forms** from the menu. You will see a list of any existing forms. Click the **New Form** button to display the designer.



**Form name** Enter a name to be used as a reference

### Form properties

- Group to add user to** The usergroup the user will be added to
- Expiry (days)** Number of days for membership (0 for no expiry)
- Thankyou page** The page to send the user to after they register. If the page is in the same folder just enter the filename. If not use a full path like [/members/registerthanks.php](#). See the section 'Sending registration data to the thank you page' on page 79 for details about displaying the entered data on the thank you page and how to automatically login the user if needed.
- Initial status** Normally a user account would be enabled after registration but you can choose to keep it disabled until you approve the request for example.
- User email** Select the email template to send to the user.
- Admin email** Select the email template to send to the admin.

### Paypal and Stripe options

If you have installed the Paypal plugin (V1.8 or above) or Stripe plugin you will also have some further options. See the plugin manuals for further details.



## Form style

The settings in this section define the general look of the form and can be set to match your site. The form is responsive (assuming your page is of course) so you can set the max form width to a suitable size. The background color setting is just used to set the background of the form preview. When the form is generated it will be transparent so the background of your site will be seen.

## Button style

In this section you can style the form button.

## Selected field properties

You can edit an existing form field by clicking the  icon next to it or click the  icon at the bottom of the form to add a new field. You can then set the properties for the field such as the Sitelok field it represents (name, email, custom1 etc) and the type of input such as text, dropdown etc (only types suitable for the field are shown). You can also set if the field will be required and for custom fields any validation required.

Most of the settings are self explanatory but its worth noting that when using the drop down menu you can set a list of options one per line like the first menu below.



If you want the displayed option and the value sent to the form to be different then enter [display,value](#) as in the second menu above. If you want the first value to be something that can't be selected like 'Select your country' then set the field as required.

The text area input type is available only for custom fields set to store more than the default 255 characters. You can change the size of custom fields in [Tools - Configuration - Custom fields](#).

### File uploads

Sitelok can use custom fields to store the filename of an uploaded file which is stored in the folder set in [Tools - Configuration - Full file path to download folder](#). So for example if you allow a user to upload a photo to the custom1 field you could then display that image on his members pages using something like

```

```

To add an upload field to a the form just add a custom field and select the [File upload](#) input type.

Uploaded files are prefixed with a timestamp to ensure they are unique so it doesn't matter if several users upload a file with the same name. The files are secure so users can not access any files that belong to other users.


For security, by default, Sitelok only allows image files of type .jpg .gif and .png to be uploaded. If you wish to create your own list of allowed types then you can add them to your slconfig.php

```
$sl_alloweduploads[]=".jpg";  
$sl_alloweduploads[]=".pdf";  
$sl_alloweduploads[]=".gif";  
etc
```

### Allow user selection of usergroup

Normally you would set the usergroup the user will join in the [Form properties](#) section. If, instead, you want to allow the user to choose a usergroup then set the default usergroup in [Form properties](#) and add a [Usergroup](#) field to your form (in [Sitelok field](#)) which will be a drop down menu. In the [Options](#) field list the usergroups they can choose from. See the notes about using the drop down menu above for more details.

### Generating the form code

Once you have the form looking as you want click [Save Changes](#) and then click [Return to form list](#). Click the  icon next to the form in the listing to generate the code to copy and paste in your page in the 3 steps shown. There are two ways to add the form to your page.

**Embedded** This links the form into your page. If you make any changes to the form in the designer they will be reflected on your page when you save them. For most situations this is the easiest way.

**Source code** If you prefer to generate full source code for the CSS, Javascript and HTML that you can further edit then choose this method.

By default the code generated will use the styling you set in the form designer which is usually what is needed. However you can set the **Simplified form styles** option to produce a simpler form that may be able to take styling from your page if it has styles set for form input elements.

### **Sending registration data to the thank you page**

After a user has successfully registered they are sent to the page defined which usually will thank them for registering. Normally it is best to email the user their assigned username and password for security reasons. However if you wish you can pass to the page any data entered in the registration form for use in that page via the URL. To do this you can use the following variables in the thank you page URL.

<b>!!!username!!!</b>	The username assigned
<b>!!!password!!!</b>	The password assigned ( <b>not recommended</b> )
<b>!!!passwordclue!!!</b>	The last three characters of the password
<b>!!!passwordhash!!!</b>	A secure hash of the password
<b>!!!name!!!</b>	The name entered
<b>!!!email!!!</b>	The email address entered
<b>!!!custom1!!!</b>	Custom1 data (use custom1 to custom50).

For example

```
thanks.php?username=!!!username!!!&name=!!!name!!!
```

This would send the username and name from the registration form to the successful registration page thanks.php. You could then display the data using PHP like this

```
<?php echo htmlentities(urldecode($_GET['username']),ENT_QUOTES,'UTF-8'); ?
```

## Auto login after registration

You can auto login a user after registration instead of sending them to a thank you page. To do this set the thank you page URL in the registeruser() function as follows:-

</page.php?username=!!!username!!!&password=!!!passwordhash!!!>

where page.php is a secure members page. Notice we use the password hash here for security. The password hash used here is secure and only valid for the registration.

## Admin approval of users before access

By default after a user has registered they will be able to login. If you wish to review users before providing access then you can set that up as follows.

- 1) In the Form properties set the **Initial account status** to **Disabled**.
- 2) In **Form properties** set the user email template to be **pending.htm** and the admin email template as **pendingadmin,htm**. These are just example emails so you can modify them or use your own. pendingadmin.htm has an approval link in it which you can click to approve (enable) a user.
- 3) On your thankyou page explain to the user that you need to approve their access before they can login.

The approval link in the example pendingadmin.htm also sets the email templates that are sent to the user and admin when the link is clicked. These are set to the examples **approve.htm** and **approveadmin.htm**.

You can add an approval link to an admin email template like this.


Open the template in the email section of the control panel. Type the text you want to click and select it. Then choose **Approve/confirm account link** in the **Insert variable** menu.

## Chapter 7 - User Registration Page

---

Visual Editor  HTML Source code

Insert Variables ▾

- verify email link
- Auto Login link**
- Auto Login link...
- Download link**
- Download link...
- File size...
- Account processing**
- Approve/confirm account link... 
- Disable account link...
- Delete account link...
- Contact form**
- Contact form data

Set the expiry minutes for the link as needed (0 for no expiry) and the email template file names for the emails to be sent to the user and admin when the link is clicked. There is no need to set a redirect page here.

Approve/confirm account link ✕

---

**Link valid for (minutes)**

Set to 0 for no expiry

**User email template (blank for none)**

**Admin email template (blank for none)**

**Redirect page URL**

Cancel

Insert

## User confirmation of email address before access

By default after a user has registered they will be able to login. If you wish to have users confirm their email first then there are two ways. The easiest is to have Sitelok assign a random password that is emailed to the user. This way they can only get the password if the email is valid.

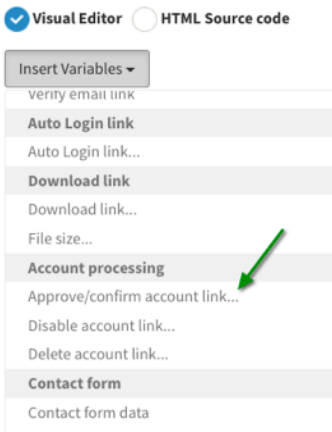
If you prefer you can have users click a confirmation link in the email to enable their account. This can be set up as follows.

- 1) In the Form properties set the **Initial account status** to **Disabled**.
- 2) In **Form properties** set the user email template to be **confirm.htm** and the admin email template as **confirmadmin.htm**. These are just example emails so you can modify them or use your own. **confirm.htm** has an approval link in it which the user clicks to confirm (enable) their account.
- 3) On your thankyou page explain to the user that they need to confirm their email before they can login.

The confirmation link in the example confirm also sets the email template that is sent to the admin when the link is clicked. This is set to the examples **confirmedemailadmin.htm**.

You can add a confirmation link to your own email template like this.

Open the template in the email section of the control panel. Type the text you want to click and select it. Then choose **Approve/confirm account link** in the **Insert variable** menu.



## Chapter 7 - User Registration Page

---

Set the expiry minutes for the link as needed keeping it as short as practical (e.g. 1440 minutes for 24 hours) and the email template file names for the emails to be sent to the user and admin when the link is clicked. You can also enter the URL to a thankyou page where the user is sent after they click the link.

Approve/confirm account link ×

---

**Link valid for (minutes)**

Set to 0 for no expiry

**User email template (blank for none)**

**Admin email template (blank for none)**

**Redirect page URL**

Cancel

Insert

## Allowing users to register again

Normally users can only register once in Sitelok. However sometimes you may want an existing client to be able to register again when their access has expired. You can allow this by adding `$sldeleteexisting=1;` to the snippet at the top of the registration page like this

```
<?php
$sldeleteexisting=1;
require_once("slpw/sitelokregister.php");
?>
<?php require_once("slpw/slregisterform.php"); ?>
```

This will only allow a user whose access has expired (no usergroups unexpired) to register again.

## Creating a registration form using your own html

For most users the form designer is the easiest and quickest way to make a registration form. If you want more control over styling etc and you have some html knowledge then you can make your own forms as described here.

For reference in this chapter please open register.php in the [html source examples](#) folder from the zip file. You can also see registerturing.php for an example using the CAPTCHA code.

The registration form you will use is just a standard html form that can be created by your page editor software. We will give the form a specific name and use certain input field names to get the data entered. Here is an example registration form which we will go on to explain in detail.

```
<?php require_once("slpw/sitelokregister.php"); ?>
<html>
<head><title>Register</title></head>
<body>
<?php if ($registermsg!="") print $registermsg; ?><br>
<form name="sitelokregisteruser" action="register.php" method="POST">
<?php
registeruser("CLIENT", "365", "thanks.php", "newuser.htm", "admin.htm", "Yes", "YNYYY"); ?>
Name <input type="text" name="name" value="<?php echo $name; ?>"><br>
Email <input type="text" name="email" value="<?php echo $email; ?>"><br>
Custom 1 <input type="text" name="custom1" value="<?php echo $custom1; ?>"><br>
```

## Chapter 7 - User Registration Page

---

```
<input type="submit" name="Button1" value="Register">
</form>
</body></html>
```

```
<?php require_once("slpw/sitelokregister.php"); ?>
```

This line is required at the very top of the page to let Sitelok know that the page contains a registration form. Change the path used to sitelokregister.php as appropriate.

```
<?php if ($registermsg!="") print $registermsg; ?>
```

This line will display any error message from Sitelok when the form is submitted. You can place this line anywhere on the page and style it as required using html. Sitelok will check that the required fields are entered and that email addresses are valid. You can define which custom fields are required in the Sitelok Configuration menu. The form is redisplayed if fields are not entered correctly.

### **registeruser() function**

The `registeruser()` function call in the form tells Sitelok how to handle the registration of the user. The function call has 7 parameters and looks like this (split on several lines for clarity).

```
<?php registeruser("CLIENT","365","/
thanks.php","newuser.htm","admin.htm","Yes","YNYYY"); ?
>
```

**CLIENT** This first parameter is the usergroup you are allowing the user to join. If you list multiple groups separated by commas then the user will be added to them all. e.g.  
`"GROUP1,GROUP2,GROUP3"`

**365** The second parameter defines the expiry time in days for the membership. Use 0 for no expiry time. You can also use an 6 digit fixed expiry date in the format you selected. If you are adding the user to multiple groups then you can list several expiry times like this.  
`"7,31,365"`

`"/thanks.php"` The third parameter is the page the user gets redirected to if successfully registered. You can send user entered

data to this thank you page if you wish. See the section titled 'Sending registration data to the thank you page'.

"newuser.htm"

The fourth parameter is the name of the email template sent to the user. The email template must exist in the slpw/email folder."

"admin.htm"

The fifth parameter is the name of the email template sent to the site administrator. The email template must exist in the slpw/email folder.

"Yes"

The sixth parameters determines if the user is enabled immediately upon registration. Set to "Yes" to allow immediate access. Set to "No" if you want the user disabled until you manually approve the registration. If you set to No you can add the **!!!approve!!!** variable to the admin email template to one-click enable the user (see the email section of the manual).

"YNYYY"

The seventh parameter provides security for the form. HTML forms are normally very easy for hackers to modify so Sitelok uses secure hashes to ensure that only the fields that you have included in the form can be accepted. Each registration field can be allowed with Y or not allowed with N like this

Character 1 Username  
Character 2 Password  
Character 3 Name  
Character 4 Email  
Character 5 Custom field 1  
Character 6 Custom field 2  
etc etc

If you include a field in the form then set the corresponding Y in the parameter. Our example form has "YNYYY" because we have used username, name, email and custom1 in the form but not the password. The string of characters can be up to 54 characters long.

## Registration form fields

The following form fields can be used in the form to collect data from the user.

**username** This field collects the username that the user has chosen to use. It is an optional field and if not included Sitelok will use the value entered in the email field as the username. Max length is 100 characters. You can display already entered information if the form is submitted unsuccessfully using the value in \$username. Here is an example.

```
<input type="text" name="username" value="<?php echo $username; ?>">
```

**password** This field collect the password that the user has chosen to use. It is an optional field and if not included Sitelok will create a random password automatically. Max length is 50 characters.

```
<input type="password" name="password">
```

**verifypassword** This field forces the user to enter a chosen password a second time for verification. It is an optional field but if included Sitelok will verify that the contents match that entered in the password field.

```
<input type="password" name="verifypassword">
```

**name** This required field collects the name of the user. You can display already entered information if the form is submitted unsuccessfully using the value in \$name. Max length is 100 characters. Here is an example.

```
<input type="text" name="name" value="<?php echo $name; ?>">
```

**email** This required field collects the email address of the user. If the username field is not included in the form then Sitelok will use this email address as the username as well. You can display already entered information if the form is submitted unsuccessfully using the value in \$email. Max length 100 characters. Here is an example.

```
<input type="text" name="email" value="<?php echo $email; ?>">
```

**verifyemail** This optional field can be used to verify a users email address. Here is an example.

```
<input type="text" name="verifyemail" value="<?php echo $verifyemail; ?>">
```

**custom1** The custom1 to custom50 fields collect extra data from the user. The fields are optional. Please note that even if you re-title a custom field it is still referred to as \$custom1 in the registration form. You can also use drop down menu (select) fields, checkboxes or radio buttons custom fields if required. Max length is 255 characters although this can be increased (see our knowledge base).

```
<input type="text" name="custom1" value="<?php echo $custom1; ?>">
```

**usergroup** See section below 'Allow user selection of usergroup'.

### **CAPTCHA code**

*CAPTCHA - Completely Automated Public Turing to tell Computers from Humans Apart*

To stop repeated automated user registration attempts you can add a CAPTCHA (turing) code field to the registration form. This forces the user to enter a series of characters displayed as a graphic to complete the process. As these characters change each time and because they are displayed in graphic form it helps to ensure that only real users rather than automated systems are filling in the form.



To enable this add the following to your form

```
<input type="text" name="turing" value="">  

```

making sure the path to turingimage.php is correct and then in the Sitelok administration configuration page enable Turing Register.

Register CAPTCHA

For an example see `registerturing.php` in the `members` folder.

### Allow user selection of usergroup

Normally you would securely set the `usergroup` to be joined in the `registeruser()` function so that users cannot join other groups. If you do however want to allow the user to select one of several groups then you can use the `usergroup` field. To control which groups could be sent in the form you should list the allowed groups in the `registeruser()` function and have these for selection in the `usergroupfield`. So for example to allow the user to select between BRONZE, SILVER and GOLD groups you would use the following `registeruser()` function call

```
<?php registeruser("BRONZE,SILVER,GOLD","365","/
thanks.php","newuser.htm","admin.htm","Yes","YNYYYY"); ?
>
```

and the `usergroup` field in the form like this

```
<select name="usergroup">
<option value="BRONZE">Bronze group</option>
<option value="SILVER">Silver group</option>
<option value="GOLD">Gold group</option>
</select>
```

In this case the user can select one of the groups which would be joined for 365 days. If you want to have each group with a different expiry time you can use this example

```
<?php registeruser("WEEK,MONTH,YEAR","7,31,365","/
thanks.php","newuser.htm","admin.htm","Yes","YNYYYY"); ?>
```

Note that when using the `usergroup` field in the registration form the user can only select one `usergroup` to join.

### File uploads

Sitelok can use custom fields to store the filenames of uploaded files which are stored in the folder pointed to by the File Location path set in Sitelok configuration. So for example if you allow a user to upload a photo to the

custom1 field you can then display that image on his members pages using something like

```

```

To add an upload field to a the user profile page use a file html input type on the form like this

```
<input type="file" name="custom1">
```

This would upload the selected file to the Sitelok file folder and set the custom1 field to the filename. Any existing file stored in the custom1 field would be deleted. Uploaded files are prefixed with a timestamp to ensure they are unique so it doesn't matter if several users uploaded a file called myphoto.jpg for example. The files are secure so users can not access any files that belong to other users.

**Important** Remember that forms that upload need

```
enctype="multipart/form-data"
```

in the form tag. So for example

```
<form name="sitelokregisteruser" action="register.php" method="POST"
enctype="multipart/form-data" >
```

If you prefer to keep upload files in a separate folder just create a folder called **profile** inside the main File Location folder. If found then Sitelok will automatically use this profile folder.

For security by default Sitelok only allows image files of type .jpg .gif and .png to be uploaded. If you wish to create your own list of allowed types then you can add this to your slconfig.php

```
$sl_alloweduploads[]=".jpg";
$sl_alloweduploads[]=".pdf";
$sl_alloweduploads[]=".gif";
etc
```

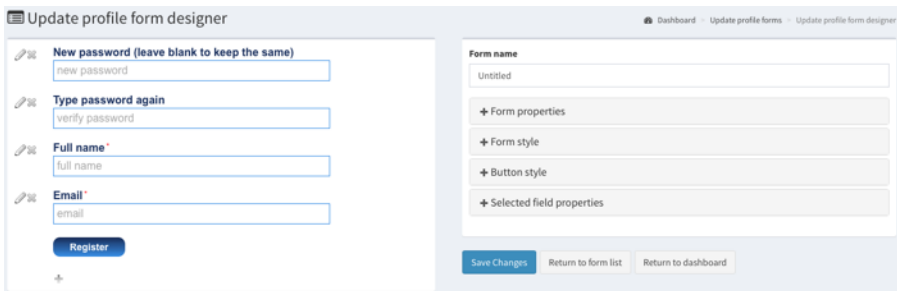
# Chapter 8 User Profile page

If you wish you can allow users to change their registration details including password, name, email address and any of the custom fields. You can also choose to allow them to change their username if you wish. This is done inside the members only area ensuring that only the real user can change this data.

There are two ways you can make update profile forms. The easiest is to use the Update profile form code feature in the admin control panel. This will generate the code needed so that you can copy and paste it in your page. If you want more flexibility you can write your own html form. Both are described in detail here.

## Creating an update profile form using the Form Designer

To open the form designer select **Forms - Update profile forms** from the menu menu. You will see a list of any existing forms. Click the **New Form** button to display the designer.



**Form name** Enter a name to be used as a reference

### Form properties

**Thankyou page** Page to send user to after they update. You can leave it blank to leave the user on the same page if you wish. In that case a message saying 'Your account has been updated' is displayed.

**User email** Select an email template to send to the user if you wish.  
**Admin email** Select the email template to send to the admin if needed.



## Form Style

The settings in this section define the general look of the form and can be set to match your site. The form is responsive (assuming your page is of course) so you can set the max form width to a suitable size. The background color setting is just used to set the background of the form preview. When the form is generated it will be transparent so the background of your site will be seen.

### Button style

In this section you can style the form button.

### Selected field properties

You can edit an existing form field by clicking the  icon next to it or click the  icon at the bottom of the form to add a new field. You can then set the properties for the field such as the Sitelok field it represents (name, email, custom1 etc) and the type of input such as text, dropdown etc (only types suitable for the field are shown). You can also set if the field will be required and for custom fields any validation required.

Most of the settings are self explanatory but its worth noting that when using the drop down menu you can set a list of options one per line like the first menu below.



If you want the displayed option and the value sent to the form to be different then enter [display,value](#) as in the second menu above. If you want the first value to be something that can't be selected like 'Select your country' then set the field as required.

The text area input type is available only for custom fields set to store more than the default 255 characters. You can change the size of custom fields in [Tools - Configuration - Custom fields](#).

### File uploads

Sitelok can use custom fields to store the filename of an uploaded file which is stored in the folder set in [Tools - Configuration - Full file path to download folder](#). So for example if you allow a user to upload a photo to the custom1 field you could then display that image on his members pages using something like

```

```


To add an upload field to a the form just add a custom field and select the [File upload](#) input type.

Uploaded files are prefixed with a timestamp to ensure they are unique so it doesn't matter if several users upload a file with the same name. The files are secure so users can not access any files that belong to other users.

For security, by default, Sitelok only allows image files of type .jpg .gif and .png to be uploaded. If you wish to create your own list of allowed types then you can add them to your slconfig.php

```
$sl_alloweduploads[]=".jpg";  
$sl_alloweduploads[]=".pdf";  
$sl_alloweduploads[]=".gif";  
etc
```

### Generating the form code

Once you have the form looking as you want click [Save](#) and then click [Return to form list](#). Click the  icon next to the form in the listing to generate the code to copy and paste in your page in the 3 steps shown. There are two ways to add the form to your page.

**Embedded** This links the form into your page. If you make any changes to the form in the designer they will be reflected on your page when you save them. For most situations this is the easiest way.

**Source code** If you prefer to generate full source code for the CSS, Javascript and HTML that you can further edit then choose this method.

By default the code generated will use the styling you set in the form designer which is usually what is needed. However you can set the [Simplified form styles](#) option to produce a simpler form that may be able to take styling from your page if it has styles set for form input elements.

## Requiring existing password to make changes

On some sites you may want the user to enter their existing password again (even though they are already logged in) before changes are accepted. Sitelok can be set to require the existing password to allow any changes or just when the user tries to change the password. To enable this

- 1) Select **Tools - Configuration** from the menu and open the **Passwords** panel to see the **Profile update password required** setting.

### Profile update password required

password required to update profile

Select either **Required for any changes** or **Required to change password**.

- 2) Add a field to the form for the user to enter their existing password by choosing **Existing password** from the Sitelok filed drop down in the form designer.

## Requiring confirmation of email change

By default Sitelok allows a user to change their email to any valid email address. However in some applications you may like to ensure that the user does own that email address by having a confirmation link sent to it. The email address will only be updated when the link is clicked in that case.

To enable this feature select **Tools - Configuration** from the menu and open the **General settings** panel. Set **Email change verification** to **Yes** and choose the email template to be sent to the user.

### Email change verification

setting yes requires users to confirm email address via emailed link

[verifyemail.htm](#) 

Once enabled whenever the email address field is updated the selected email template will be sent to the requested email address. When the link is clicked the email address will be updated. The standard email template defined for the form will still be sent to the existing email address when the form is first submitted as confirmation.

If you prefer to have the confirmation link sent only to the existing email address instead then you can set the email template to 'No email' in the configuration page and include the link in the normal template defined for the form.

In email template you can use these template variables.

**!!!verifyemail!!!** This inserts a link that when clicked will update the email address. This should normally be used in a hyperlink like this

```
<a href="!!!verifyemail!!!">click to confirm</a>
```

**!!!requestedemail!!!** The requested password.

There is an example email template called verifyemail.htm.

### Creating a profile update form using your own html

For most users the form designer is the easiest and quickest way to make an update form. If you want more control over styling etc and you have some html knowledge then you can make your own forms as described here.

For reference in this chapter please open update.php in the [html source examples](#) folder from the zip file.

The profile update form you will use is just a standard html form that can be created by your page editor software. We will give the form a specific name and use certain input field names to get the data entered. Here is an example profile form which we will go on to explain in detail.

```
<?php require_once("slpw/sitelokpw.php"); ?>
<html>
<head><title>Update registration details</title></head>
<body>
<?php if ($msg!="") print $msg; ?><br>
<form name="sitelokmodify" method="post" action="<?php print
$thispage; ?>" onSubmit="return validateprofile()">
<?php sitelokmodify("updateuser.htm", "updateuseradmin.htm", "/"
success.php", "NYYYY"); ?>
New Password (leave blank to not change it) <input type="text"
name="newpassword"><br>
Verify Password <input type="text" name="verifynewpassword"><br>
```

## Chapter 8 - User Profile Page

---

```
Name <input type="text" name="newname" value="<?php echo $slname; ?>"><br>
Email <input type="text" name="newemail" value="<?php echo $slemail; ?>"><br><br>
Custom1 <input type="text" name="newcustom1" value="<?php echo $slcustom1; ?>"><br>
<input type="submit" value="Save Changes">
</form>
</body>
</html>
```

```
<?php require_once("slpw/sitelokpw.php"); ?>
```

This line is required secures the page for access by members only. As no usergroups are listed all groups can access this form. Change the path used to sitelokpw.php as appropriate.

```
<?php if ($msg!="") print $msg; ?>
```

This line will display any error message from Sitelok when the form is submitted. You can place this line anywhere on the page and style it as required using html. Sitelok will check that the required fields are entered and that email addresses are valid. You can set custom fields as required in the admin Configuration menu.

### **sitelokmodify() function**

The sitelokmodify() function call in the form tells Sitelok how to handle the data sent from the form. The function call has 4 parameters and looks like this (split on several lines for clarity).

```
<?php sitelokmodify("updateuser.htm","updateuseradmin.htm",
"/success.php","NYYYY"); ?>
```

"updateuser.htm"

The first parameter is the email template to send to user when they update their profile. The template should be in the slpw/email folder. Set to "" to not send an email.

"updateuseradmin.htm"

The second parameter is the email template to send to the administrator when a user updates their profile. The template should be in the slpw/email folder. Set to "" to not send an email.

`"/success.php"`

You can redirect the user to a specified page after successfully updating their profile if you wish.

`"NYYYY"`

The fourth parameter provides security for the form. HTML forms are normally very easy for hackers to modify so Sitelok uses secure hashing to ensure that only the fields that you have included in the form can be accepted. Each registration field can be allowed with Y or not allowed with N like this:-

Character 1 Username  
Character 2 Password  
Character 3 Name  
Character 4 Email  
Character 5 Custom field 1  
Character 6 Custom field 2  
etc etc

If you include a field in the form then set the corresponding Y in the parameter. Our example form has "NYYYY" because we have used password, name, email and custom1 in the form but not the username. The string of characters can be up to 54 characters long.

### Modify Profile form fields

The following form fields can be used in the form allowing the user to modify their stored profile data.

`newusername`

This field allows the username to be modified. In most cases it is usually best not to allow users to change their usernames once they have joined. You can prefill the username using the value in `$newusername`. Max length is 100 characters. Here is an example.

```
<input type="text" name="newusername" value="<?php echo $newusername; ?>">
```

`newpassword`

This field allows the user to change their password. Max length 50 characters. Here is an example.

```
<input type="password" name="newpassword">
```

**verifynewpassword** This field forces the user to enter a chosen password a second time for verification. It is an optional field but if included Sitelok will verify that the contents match that entered in the newpassword field.

```
<input type="password" name="verifynewpassword">
```

**newname** This field allows the user to change their stored name. You can prefill the name if the form is submitted unsuccessfully using the value in \$newname. Max length 100 characters. Here is an example.

```
<input type="text" name="newname" value="<?php echo $newname; ?>">
```

**newemail** This field allows the user to change their stored email address. You can prefill the entered email using the value in \$newemail. Here is an example.

```
<input type="text" name="newemail" value="<?php echo $newemail; ?>">
```

**newcustom1** The newcustom1 to newcustom50 fields collect extra data from the user. Note that even if you re-title a custom field it is still referred to with \$newcustom1 in the modify profile form. You can also use drop down menu (select) fields, checkboxes or radio buttons custom fields if required. Max length 255 characters although this can be increased (see our knowledge base).

```
<input type="text" name="newcustom1" value="<?php echo $newcustom1; ?>">
```

**existingpassword** If you have enabled **Profile update password required** then this field is used to get the password from the user.

The rest of this chapter discusses features that are optional so you can skip it if you wish.

### Syncing email and username

Normally the email field on the modify profile form will only update the email address of the user. To have the email field also update the username field automatically you need to do the following in the form.

- 1) Include the email field in the form but not a username field.
- 2) Set the fourth security parameter of the `sitelokmodify()` function to allow both the email field and the username field (even though the username field is not in the form). For example

```
"YYYYY"
```

Now when a user enters a new email address it will change the username as well (checking it is not already used).

### Allowing file uploads

You may like to allow the user to upload one or more files such as a user photo etc. Sitelok can use custom fields to store a filename of an uploaded file which are stored in the folder pointed to by the File Location path set in Sitelok configuration. So for example if you allow a user to upload a photo to the custom1 field you can then display that image on his members pages using something like

```

```

To add an upload field to a the user profile page use a file html input type on the form like this

```
<input type="file" name="newcustom1">
```

This would upload the selected file to the Sitelok file folder and set the custom1 field to the filename. Any existing file stored in the custom1 field would be deleted. Uploaded files are prefixed with a timestamp to ensure they are unique so it doesn't matter if several users uploaded a file called myphoto.jpg for example.

**Important** Remember that forms that upload need

```
enctype="multipart/form-data"
```

in the form tag. So for example

```
<form name="sitelokregisteruser" action="register.php" method="POST"
enctype="multipart/form-data" >
```

So that the user does not have to upload the file every time the form is submitted if a file is not selected any existing file is left in place. If you want to allow the user to delete an existing upload file you can use the following.

```
<input type="checkbox" name="newcustom1clear" value="Clear">
```

If you prefer to keep upload files in a separate folder just create a folder called **profile** inside the mail File Location folder. If found then Sitelok will automatically use this profile folder.

For security by default Sitelok only allows image files of type .jpg .gif and .png to be uploaded. If you wish to create your own list of allowed types then you can add this to your slconfig.php

```
$sl_alloweduploads[]=".jpg";  
$sl_alloweduploads[]=".pdf";  
$sl_alloweduploads[]=".gif";  
etc
```

### Adding or removing groups

If you wish you can add form fields to the modify profile form that add, replace, remove or extend group membership for the user. You can do this using hidden fields or using checkboxes. The form inputs are secure and can't be tampered with and you can add as many fields as you need to the form. As you can have any number of modify profile forms on your site you could perhaps use different forms for different purposes etc.

### Add a user to a group

To add a user to a group when the modify profile form is submitted you can add the following field to the form

```
<input type="hidden" name="modifygroup[]" value="<?php  
addgroupinput("GROUP","365"); ?>">
```

or using a checkbox

```
<input type="checkbox" name="modifygroup[]" value="<?  
php addgroupinput("GROUP","365"); ?>">
```

"GROUP"

The group to add the the user to

"365"            The expiry for the group. This can be in days (e.g. 365) or a fixed six digit date. Here are some examples  
"365"            This would set the new group with an expiry 365 days from now  
"010809"        This would set the expiry to 1st August 2009 assuming DDMMYY format

### Remove a user from a group

To remove the user from a group when the modify profile form is submitted you can add the following field to the form

```
<input type="hidden" name="modifygroup[]" value="<?php removegroupinput("GROUP"); ?>">
```

or using a checkbox

```
<input type="checkbox" name="modifygroup[]" value="<?php removegroupinput("GROUP"); ?>">
```

"GROUP"            The group to remove

### Replace a group

To replace a group with another when the modify profile form is submitted you can add the following field to the form

```
<input type="hidden" name="modifygroup[]" value="<?php replacegroupinput("GROUP", "NEWGROUP", "365"); ?>">
```

or using a checkbox

```
<input type="checkbox" name="modifygroup[]" value="<?php replacegroupinput("GROUP", "NEWGROUP", "365"); ?>">
```

"GROUP"            The group to add the the user to  
"365"            The expiry for the group. This can be in days (e.g. 365) or a fixed six digit date. Here are some examples  
""                Keep current expiry date  
"365"            This would set the new group with an expiry 365 days from now

"010809" This would set the expiry to 1st August 2009 assuming DDMMYY format

### Extend a group

To extend membership of a group when the modify profile form is submitted you can add the following field to the form

```
<input type="hidden" name="modifygroup[]" value="<?php extendgroupinput("GROUP","365","0"); ?>">
```

or using a checkbox

```
<input type="checkbox" name="modifygroup[]" value="<?php extendgroupinput("GROUP","365","0"); ?>">
```

"GROUP"

The group to be extended

"365"

The expiry for the group. Here are some examples

"365"

This would set the new group with an expiry 365 days from now

"010809"

This would set the expiry to 1st August 2009 assuming DDMMYY format

"+30"

This would extend the current expiry by 30 days

"0"

This setting can be used to determine whether to extend the expiry. Here are some examples

"0"

Setting 0 will always adjust the expiry date

"+30"

Using a + in front of the number will only adjust the expiry date if the existing expiry was 30 (in this case) or more days ago.

"-30"

Using a - in front of the number will only adjust the expiry if there are at least 30 (in this case) days until the current expiry.

# Chapter 9 Login Template

## Login page types

There are two types of login page that Sitelok uses which we call a Login Template or a Login Page. It is important to understand the difference.

### Login Template (as described in this chapter)

The login template is the page displayed by Sitelok when a user tries to directly access a secure page (and they are not already logged in) or when you link to /slpw/login.php. A login template is **not** a page that the user visits to login. i.e. it is not a page that a user visits directly.

### Login Page

A login page is a publicly accessible page on your site that contains a user login form on it. For many sites you don't need such a login form because you can just link users directly to a secured page which causes the login template to be displayed (if the user is not already logged in). Alternatively you can point to /slpw/login.php which also displays the default login template and can redirect users based on their usergroup after login.

We will describe how to create a **Login Template** in this chapter.

## The default Login Template

The login template is the page displayed by Sitelok when a user tries to directly access a secure page (and they are not already logged in) or when you link to /slpw/login.php. By default Sitelok has a built in template that is used which looks like this.



You can change the style of the form (colors, text, background etc) in the [Tools - Default Login Form Style](#) page.

When you call `/slpw/login.php` the login template appears and after login the user is redirected to the page you specified for that usergroup. You can also force a redirect to a different page instead after login by calling it like this

```
/slpw/login.php?redirect=/folder/page.php
```

One handy use for this technique is if you want to have a login link or button that takes the user to login and then brings them back to the same page. You can either include the page in the link as above or automate it using

```
/slpw/login.php?redirect=<?php echo \$startpage; ?>
```

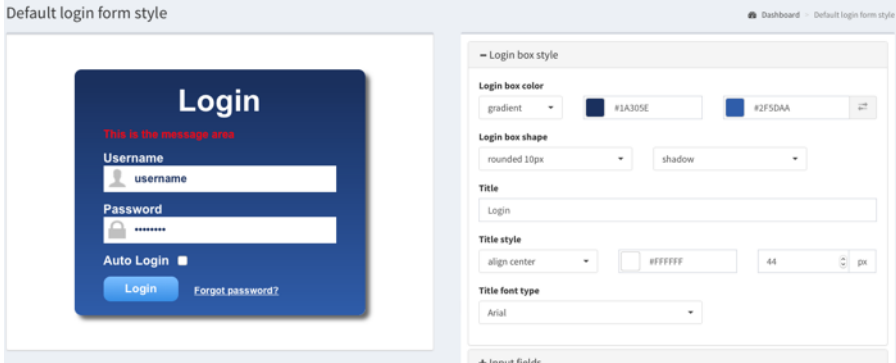
Normally you would use this on a page 'secured' using the PUBLIC usergroup for access. This allows the page to be viewed without logging in (otherwise the login button would be seen) but allowing you to have parts of the page visible only when logged in. See later in the manual for more information about that.

The default login (with some styling of your own) is probably fine for most sites but you can make your own template to use instead. There are two ways you can make the login form code to put on your template page. The easiest is to use the Login form code feature in the admin control panel. This will generate the code needed so that you can copy and paste it in your template page. If you want more flexibility you can write your own html form. Both are described in detail here.

### **Creating a login template using the Form Designer**


First design your page as required making sure that any links within the page use absolute url's rather than relative. This is because the page location as far as the browser is concerned will change so relative links could end up pointing to the wrong location. Name the page with the .php extension.

Open the form designer by selecting Forms - [Login forms](#) in the menu. You will see a list of any existing forms. Click the [New Form](#) button to display the designer.



Enter a name for the login form as a reference and style the form how you wish.

### Generating the form code

Once you have the form looking as you want click **Save** and then click **Return to form list**. Click the  icon next to the form in the listing to generate the code to copy and paste in your page in the steps shown.

There are few options to set here.

1) As you are making a login template select **I am making a login template to replace the default login template**

2) There are two ways to add the form to your page. Choose the method you prefer.

**Embedded** This links the form into your page. If you make any changes to the form in the designer they will be reflected on your page when you save them. For most situations this is the easiest way.

**Source code** If you prefer to generate full source code for the CSS, Javascript and HTML that you can further edit then choose this method.

3) By default the code generated will use the styling you set in the form designer which is usually what is needed. However you can set the **Simplified form styles** option to produce a simpler form that may be able to take styling from your page if it has styles set for form input elements.

4) Follow the steps in the page to copy and paste the form code to your template page. To use the login template upload it to the slpw folder. We now need to tell Sitelok to use this template so select **Tools - Configuration** in the menu and open the **Pages and templates** panel. In the **Login template** field set the filename of the template like this.

### Login template

logintemplate.php

**Important** Only enter the filename of the template. No path is needed.

## Creating a login template using your own html form

For most users the form designer is the easiest and quickest way to make a login form. If you want more control over styling etc and you have some html knowledge then you can make your own template as described here. For reference we have included two simple examples called `logintemplate.php` and `logintemplate.turing.php` in the **html source examples** folder from the zip file. These need to be copied to the slpw and set in the configuration to work.

- 1) Design your page as required making sure that any links within the page use absolute url's rather than relative. This is because the page location as far as the browser is concerned will change so relative links could end up pointing to the wrong location. Name the page with the `.php` extension.
- 2) Within the page create the login form you wish to use making sure that text entry fields are defined for the username and password. These form fields must be named **username** and **password** respectively.
- 3) Replace the form definition with the following:

```
<form name="siteloklogin" action="<?php print $startpage; ?>"
method="POST" onSubmit="return validatelogin()" >
<?php siteloklogin(); ?>
```

The html should be entered as is. You do not need to change any values.

- 4) You need to add to your login template an area to display a short message line. This is the message that is displayed when a user enters the wrong password for example. Just add the following line where you

want this to be displayed.

```
<p id="loginmessage"><?php if ($msg!="") echo $msg; ?></p>
```

5) To use the login template save the page with the PHP extension and upload the page to the slpw folder. We now need to tell Sitelok to use this template so select **Tools - Configuration** in the menu and open the **Pages and templates** panel. In the **Login template** field set the filename of the template like this.

**Login template**

logintemplate.php

**Important** Only enter the filename of the template. No path is needed.

Here is the html for a basic login template.

```
<html>
<head>
<title>Login</title>
</head>
<body>
<p id="loginmessage"><?php if ($msg!="") echo $msg; ?></p>
<form name="siteloklogin" action="<?php print $startpage; ?>"
method="post" onSubmit="return validatelogin()">
<?php siteloklogin(); ?>
Username <input type="text" name="username" value=""><br>
Password <input type="password" name="password">
<input type="Submit" name="login" value="Login">
</form>
</body>
</html>
```

The following sections describe optional features you can add to your login template.

### Forgot my password link

You can optionally add a feature enabling a user to be sent their password by email if they forget it. In this case the user needs to enter either their username or registered email address into the username field of the login form and click a link. You can add this link to your login template as follows:-

```
<a href="javascript: void forgotpw()">Forgot your  
password?</a>
```

### Remember login details / Auto login

If you wish to allow users to have their login details remembered then you should add a check box with the name 'remember' which the user will check when they want their login details stored in a cookie. The checkbox line html should look exactly as follows:-

```
<input name="remember" type="checkbox" value="1" <?php  
if ($slcookielogin=="1") print "CHECKED"; ?>>
```

So that the username and password are entered into the relevant form fields when necessary you should also add the value tag to the username and password fields as follows.

```
value="<?php print $slcookieusername; ?>"  
value="<?php print $slcookiepassword; ?>"
```

When the checkbox is checked the users last used username and password will be filled in the form automatically.

The above can be modified to enable **automatic login** instead if preferred by changing the check box value from "1" to "2". In this case a user will be automatically logged in fully without seeing the login form until they logout. This will work even if you enable CAPTCHA for login while still stopping automated systems trying to guess passwords.

To use either method you need to enable it. The setting can be found by selecting **Tools - Configuration** in the menu and opening the **General settings** panel.

#### Remember me

**Remember Me** Enables stored login details  
**Auto Login** Enables auto login mode

### CAPTCHA (turing) code

*CAPTCHA - Completely Automated Public Turing to tell Computers from Humans Apart*

To stop repeated automated user login attempts you can add a CAPTCHA (turing) code field to the login form. This forces the user to enter a series of characters displayed as a graphic to complete the login. As these characters change each time and because they are displayed in graphic form it helps to ensure that only real users rather than automated systems are filling in the form.



To enable this add the following to your form

```
<input type="text" name="turing" value="">

```

making sure the path to turingimage.php is correct and then enable Captcha Login. This setting can be found by selecting **Tools - Configuration** from the menu and opening the **General settings** panel.

#### Login CAPTCHA

Enabled ▼

Here is a complete example login template with the forgot password, remember me, and turing code features included.

```
<html>
<head><title>Login</title></head>
<body>
<p id="loginmessage"><?php if ($msg!="") echo $msg; ?></p>
<form name="siteloklogin" action="<?php print $startpage; ?>"
method="post" onSubmit="return validatelogin()">
<?php siteloklogin(); ?>
Username
<input type="text" name="username" value="<?php print
$slcookieusername; ?>">
<br>
Password
<input type="password" name="password" value="<?php print
$slcookiepassword; ?>">
<br>
<input type="text" name="turing" value="">
```

## Chapter 9 - Login Template

---

```
<br>
<input type="Submit" name="login" value="Login">
<br>
<input name="remember" type="checkbox" value="1"
<?php if ($slcookielogin=="1") print "CHECKED"; ?>
Remember me <br>
<a href="javascript: void forgotpw()"
title="Forgot your password? Enter username or email & click
link">
Forgot your password?</a>
</form>
</body>
</html>
```

# Chapter 10 Login Pages

## Login page types

There are two types of login page that Sitelok uses which we call a Login Template or a Login Page. It is important to understand the difference.

### Login Template

The login template is the page displayed by Sitelok when a user tries to directly access a secure page (and they are not already logged in) or when you link to /slpw/login.php. A login template is **not** a page that the user visits to login. i.e. it is not a page that a user visits directly.

### Login Page (as described in this chapter)

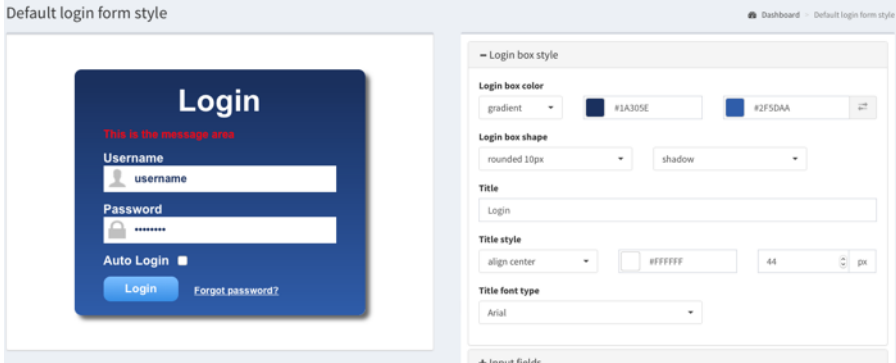
A login page is a publicly accessible page on your site that contains a user login form on it. For many sites you don't need such a login form because you can just link users directly to a secured page which causes the login template to be displayed (if the user is not already logged in). Alternatively you can point to /slpw/login.php which also displays the default login template and can redirect users based on their usergroup after login.

We will describe how to create a **Login Page** in this chapter.

There are two ways you can make the login form code to put on your page. The easiest is to use the Login form code feature in the admin control panel. This will generate the code needed so that you can copy and paste it in your page. If you want more flexibility you can write your own html form. Both are described in detail here.


## Creating a login form using the Form Designer

Open the form designer by selecting Forms - [Login forms](#) in the menu. You will see a list of any existing forms. Click the [New Form](#) button to display the designer.



Enter a name for the login form as a reference and style the form how you wish.

### Generating the form code

Once you have the form looking as you want click **Save** and then click **Return to form list**. Click the  icon next to the form in the listing to generate the code to copy and paste in your page in the steps shown.

There are few options to set here.

- 1) As you are making a login template select **I am adding the login form to a page on my site**.
- 2) There are two ways to add the form to your page. Choose the method you prefer.

**Embedded** This links the form into your page. If you make any changes to the form in the designer they will be reflected on your page when you save them. For most situations this is the easiest way.

**Source code** If you prefer to generate full source code for the CSS, Javascript and HTML that you can further edit then choose this method.

- 3) By default the code generated will use the styling you set in the form designer which is usually what is needed. However you can set the **Simplified form styles** option to produce a simpler form that may be able to take styling from your page if it has styles set for form input elements.

4) Choose if you want users to be redirected to their usergroup start page after login or simply to remain on the page that has the login form (but logged in).

5) Choose whether you want to hide the login form on the page if the user is already logged in. If you do this you could replace the form with a logout link or welcome message as described elsewhere in the manual.

Follow the rest of the steps described on the page to add the form to your page.

### Creating login form using your own html

For reference please open `pagewithlogin.php` in the [html source examples](#) folder from the zip file.

1) Within your page create the login form you wish to use making sure that text entry fields are defined for the username and password. These form fields must be named `username` and `password` respectively.

2) Replace the form definition with the following:

```
<form name="siteloklogin"
action="<?php print $startpage; ?>"
method="post"
onSubmit="return validatelogin()">
<?php siteloklogin(); ?>
```

The html should be entered as is. You do not need to change any values.

3) You need to add an area to display a short message line near to your login form. This is the message that is displayed when a user enters the wrong password for example. Just add the following line where you want this to be displayed.

```
<p id="loginmessage"><?php if ($msg!="") echo $msg; ?></p>
```

4) So that Sitelok can have control of the page with the login form on it we will secure it but using a special group called PUBLIC. This group allows anyone to visit the page while allowing Sitelok to control what happens. Add the following to the top of your page.

```
<?php
```

```
$groupswithaccess="PUBLIC";
$loginpage="pagewithlogin.php";
$logoutpage="pagewithlogin.php";
require_once("slpw/sitelokpw.php");
?>
```

You should replace `pagewithlogin.php` with the name of your page and also adjust the path to `sitelokpw.php` as required.

Here is the html for a simple login form on a page.

```
<?php
$groupswithaccess="PUBLIC";
$loginpage="pagewithlogin.php";
$logoutpage="pagewithlogin.php";
require_once("slpw/sitelokpw.php");
?>
<html>
<head><title>Page with login</title></head>
<body>
If you want to login please use the form<br><br>
<p id="loginmessage"><?php if ($msg!="") echo $msg; ?></p>
<form name="siteloklogin" action="<?php print $startpage; ?>"
method="POST" onSubmit="return validatelogin()" >
<?php siteloklogin(); ?>
Username <input type="text" name="username" value=""><br>
Password <input type="password" name="password" value="" ><br>
<input type="Submit" name="login" value="Login"><br>
</form>
</body></html>
```

The following sections describe optional features you can add to your login page.

### Forgot my password link

You can optionally add a feature enabling a user to be sent their password by email if they forget it. In this case the user needs to enter either their username or registered email address into the username field of the login form and click a link. You can add this link to your login template as follows:-

```
<a href="javascript: void forgotpw()">Forgot your password?</a>
```

## Remember login details

If you wish to allow users to have their login details remembered then you should add a check box with the name 'remember' which the user will check when they want their login details stored in a cookie. The checkbox line html should look exactly as follows:-

```
<input name="remember" type="checkbox" value="1" <?php  
if ($slcookielogin=="1") print "CHECKED"; ?>>
```

So that the username and password are entered into the relevant form fields when necessary you should also add the value tag to the username and password fields as follows.

```
value="<?php print $slcookieusername; ?>"  
value="<?php print $slcookiepassword; ?>"
```

When the checkbox is checked the users last used username and password will be filled in the form automatically.

The above can be modified to enable **automatic login** instead if preferred by changing the check box value from "1" to "2". In this case a user will be automatically logged in fully without seeing the login form until they logout. This will work even if you enable CAPTCHA for login while still stopping automated systems trying to guess passwords.

To use either method you need to enable it. The setting can be found by selecting **Tools - Configuration** in the menu and opening the **General settings** panel.

### Remember me

**Remember Me** Enables stored login details  
**Auto Login** Enables auto login mode

### Turing (CAPTCHA) code

[CAPTCHA - Completely Automated Public Turing to tell Computers from Humans Apart](#)

To stop repeated automated user login attempts you can add a CAPTCHA (turing) code field to the login form. This forces the user to enter a series of characters displayed as a graphic to complete the login. As these characters change each time and because they are displayed in graphic form it helps to ensure that only real users rather than automated systems are filling in the form.



To enable this add the following to your form

```
<input type="text" name="turing" value="">

```

making sure the path to turingimage.php is correct and then enable Captcha Login. This setting can be found by selecting [Tools - Configuration](#) from the menu and opening the [General settings](#) panel.

#### Login CAPTCHA

Enabled ▼

Here is a complete example login template with the forgot password, remember me and turing code features included.

```
<?php
$groupswithaccess="PUBLIC";
$loginpage="pagewithlogin.php";
$logoutpage="pagewithlogin.php";
require_once("slpw/sitelokpw.php");
?>
<html>
<head><title>Page with login</title></head>
<body>
If you want to login please use the form<br><br>
<?php if ($msg!="") print $msg; ?>
<form name="siteloklogin" action="<?php print $startpage; ?>"
method="POST" onSubmit="return validatelogin()">
<?php siteloklogin(); ?>
Username
```

## Chapter 10 - Login Pages

---

```
<input type="text" name="username" value="<?php print
$slcookieusername; ?>">
<br>
Password
<input type="password" name="password" value="<?php print
$slcookiepassword; ?>">
<br>
<input type="text" name="turing" value="">
<br>
<input type="Submit" name="login" value="Login"><br>
<input name="remember" type="checkbox" value="1"
<?php if ($slcookielogin=="1") print "CHECKED"; ?>>
Remember me <br>
<a href="javascript: void forgotpw()"
title="Forgot your password? Enter username or email & click
link">
Forgot your password?</a>
</form>
</body></html>
```

### Redirect upon login

If you have defined usergroups in the Manage Groups page of Sitelok Admin then you can have users redirect to specified pages when they login. The page can be defined for all users for that group or can be a page specified in one of the users custom fields. See the Sitelok Control Panel chapter for further details.

To enable login redirection on a login page add this line to the code snippet at the top.

```
$loginredirect=2;
```

like this

```
<?php
$groupswithaccess="PUBLIC";
$loginpage="pagewithlogin.php";
$logoutpage="pagewithlogin.php";
$loginredirect=2;
require_once("slpw/sitelokpw.php");
?>
```

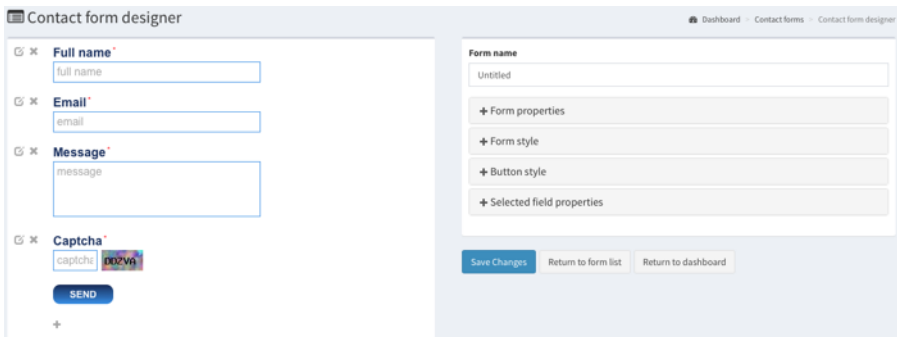
This will cause a user to be redirected whenever they login.

# Chapter 11 Contact Forms

Although most site editors include a simple contact form system we decided to include one in Sitelok so you can keep the style of your forms the same. It also allows you to integrate the contact form with Sitelok so that you can show or hide fields depending on whether the user is logged in as a member or just a visitor. You can also pre-fill fields with data from a users Sitelok data.

## Creating a contact form using the Form Designer

To open the form designer select **Forms - Contact forms** from the menu. You will see a list of any existing forms. Click the **New Form** button to display the designer.



**Form name** Enter a name to be used as a reference

### Form properties

**Send to** The email address(es) to send the contact form details to. List one per line.

**Thankyou page** Page to send the user to after they submit the form successfully. If the page is in the same folder just enter the filename. If not use a full path like </members/contactthanks.php>

**User email (visitor)** Select the email template to send (if they are a visitor) to the user when they submit the form. If you don't want to send the user an email select **None**. You can create and edit email templates in the Email section of the control panel. There are some special tags that can be used in the email templates used with the contact form.

**!!!contactform!!!**

This will insert the contact form details submitted

**!!!contactlabel\_0!!!**

This will insert the label of the first field in the form

**!!!contactvalue\_0!!!**

This will insert the entered value of the first field in the form

Replace the 0 in the above tags with 1 for the second field, 2 for the next etc.

**Admin email (visitor)** Select the email template to send to the admin when a visitor submits the contact form.

**User email (member)** Select the email template to send to the user when a logged in member submits the contact form.

**Admin email (member)** Select the email template to send to the admin when a logged in member submits the contact form.

**From Name for user** Enter the from name the user will see as the email sender. By default this is the Site Name

**Reply to address** Set the email address to receive any replies from the user email. By default this is the Site Admin email.

**Send email as user** If you set this to **No** then Sitelok will send the email to you using the Site Email as the sender. The users name will be used as the from name and their email address as the reply-to. This means you can still click reply to respond to them. This is the default method.

If you set **Yes** then Sitelok will send the contact email to you as if it was sent directly from the user. This

has the advantage that the email you see will be from the user so their name appears in your email client. You can click reply to to reply. Your hosting server or email server may not allow this though as it is seen as spoofing the email sender. Also if you use Amazon SES you cannot use this feature.

**Max attachment size** Set the maximum attachment size you allow to be sent if you use a file attachment field in the form. This should be in bytes. By default this is 2000000.

**Attachment types** List the file types you will allow as attachments. Separate each by a space. By default this is set to .jpg .gif .png .zip .txt .pdf


### Form Style

The settings in this section define the general look of the form and can be set to match your site. The form is responsive (assuming your page is of course) so you can set the max form width to a suitable size. The background color setting is just used to set the background of the form preview. When the form is generated it will be transparent so the background of your site will be seen.

### Button style

In this section you can style the form button.

### Selected field properties

You can edit an existing form field by clicking the  icon next to it. You can then set the properties for the field such as the type of input (text, dropdown etc). You can also set if the field will be required and any validation required. Most of the settings are self explanatory but a couple of points are worth mentioning.

### Drop down menus

When using the drop down menu you can set a list of options one per line like the first menu below.



If you want the displayed option and the value sent to the form to be different then enter **display,value** like as in the second menu above.

### Prefill field using (for members)

When a user is logged in as a member you can have the field pre-filled with the users Sitelok info. For example you could fill in the email and name field to save them typing those.

### In admin email header use as

When sending the contact email to you Sitelok needs to know which fields to use for the reply to email (the users email address) and the from name. Select from **name** or **reply-to email** for the relevant fields in your form. If the form is for use by logged in members only and the fields are not defined Sitelok will use the members stored name and email address by default.


### Include field for

For each field in the form you can decide whether this is included for all users, only visitors or only for logged in members. This enables you, for example, to have a captcha field for visitors but not for logged in members.

## Adding a new field

To add a new field to the form click the **+** icon at the bottom of the form.

## Generating the form code

Once you have the form looking as you want click **Save** and then click **Return to form list**. Click the  icon next to the form in the listing to generate the code to copy and paste in your page in the 3 steps shown.

There are two ways to add the form to your page.

**Embedded** This links the form into your page. If you make any changes to the form in the designer they will be reflected on your page when you save them. For most situations this is the easiest way.

**Source code** If you prefer to generate full source code for the CSS, Javascript and HTML that you can further edit then choose this method.

By default the code generated will use the styling you set in the form designer which is usually what is needed. However you can set the **Simplified form styles** option to produce a simpler form that may be able to take styling from your page if it has styles set for form input elements.

### **Sending contact form data to the thank you page**

After a user has successfully submitted the form they are sent to the thank you page. If you have defined a user email template they will be email that too. If you wish to send the email address or name entered by the user to the thank you page (so you can display it) then you can add them to the thank you page setting in the form designer like this

[contactthanks.php?email=!!!email!!!&name=!!!name!!!](mailto:contactthanks.php?email=!!!email!!!&name=!!!name!!!)

On the thank you page you can then display them using

```
<?php
echo htmlentities(urldecode($_GET['email']), ENT_QUOTES, 'UTF-8');
?>
```

```
<?php
echo htmlentities(urldecode($_GET['name']), ENT_QUOTES, 'UTF-8');
?>
```

## Chapter 12 User messages & warnings

There are several default messages used by Sitelok that are displayed to warn users about problems etc. These can be replaced with your own pages.

### Message and error page

By default when a message or error needs to be displayed to a user it is shown as a text message on a blank page. You can use your own page to display messages though. Just ensure the page has the extension .php and set the URL in the **Message and error template** setting in **Tools - Configuration - Pages and templates**. For example

<http://www.yoursite.com/messages.php>

Insert the following code in the page where you wish to display the message.

```
<?php
echo htmlentities(urldecode($_GET['slmessage']), ENT_QUOTES, 'UTF-8');
?>
```

If you need to you can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to sitelokpw.php.

```
$messagepage="http://www.yoursite.com/messages.php";
```

### Expired membership page URL

By default when a users membership to a particular user group has expired the message page is displayed with the text

[Access to this page is blocked because your membership has expired](#)

However you can create a custom page for this purpose that could perhaps contain a buy button enabling them to renew membership etc. Enter the full URL to the required page in the **Expired membership page URL** setting in **Tools - Configuration - Pages and templates**. For example

<http://www.yoursite.com/expired.php>

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$expiredpage="http://www.yoursite.com/expired.php";
```

You can also allow access to certain secured pages even when the users membership has expired by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$allowexpireduser="Y";
```

This is useful on pages where you still want to ensure the users has logged in but the fact that their membership has expired is not important. A good example is on an expired page itself.

### Wrong group page

Normally when a user is logged in but tries to access a page that is accessible only to other user groups the message page displays the following text

Your membership does not allow access to this page

You can create a custom page for this purpose if you prefer. Enter the full URL to the required page in the **Wrong group page** setting in **Tools - Configuration - Pages and templates**. For example

```
http://www.yoursite.com/wronggroup.php
```

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$wronggroupage="http://www.yoursite.com/wronggroup.php";
```

### No access page URL

Normally when a user tries to access a secured page and they are not logged in Sitelok will display the login form. On some sites you may prefer users not to bookmark and access pages in this way. Instead you can have users redirected to the page specified in this setting. This page could be your login form perhaps or alternatively a page explaining how to join the members area. Enter the full URL to the page in the **No access page url** setting in **Tools - Configuration - Pages and templates**. For example

<http://www.yoursite.com/noaccesspage.php>

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$noaccesspage="http://www.yoursite.com/noaccesspage.php";
```

If you have this setting pointing to a login page you can optionally set a login redirect flag as well. To have the user redirect back to the page they came from add `?redirect=page` to the URL. To redirect to the usergroup start page then add `?redirect=user`. For example

<http://www.yoursite.com/noaccesspage.php?redirect=page>

## Forgotten password email template

You can select the email template to send to users when they request a forgotten password from the login form. The setting **Forgotten password email** can be found in **Tools - Configuration - Passwords** (by default this is `forgotpass.htm`).

## Translating or modifying messages

You can change the wording or translate any messages that end users see. To do this open up `slconfig.php` and look for the text you want to change. For example to change 'Please enter your password' you will modify this line.

```
define("MSG_ENTERPASS", "Please enter your password");
```

Only change the text that is shown in bold here.

If you have upgraded an older version of Sitelok you may not have these definitions in your `slconfig.php` file as your existing file was kept during the upgrade. You can copy the definitions over from the latest `slconfig.php` in the `sitelokpw.zip`. Contact us to if you need any help.

## Sites using multiple languages

If your site supports several languages using different folders or a cookie setting there is usually a way to control which set of message defines are used. As this will depend on your site setup please let us know if you need help.

By default the email template used when a user requests a forgotten password (or password reset) is set in the control panel. There may be times when you want to set that based on the language chosen. In that case you can set the template from within `slconfig.php` using

```
$forgottenemail="forgotpass.htm";
```

## Chapter 13 Public access and page content control

One of Sitelok's powerful features is the ability to control how pages are displayed depending on various factors. For example you can use Sitelok to secure a page but still allow public access to all or just parts of the page. You can have parts of a page only displayed for certain usergroups or even just certain users. Because we use PHP, handling this is simple for beginners but allows developers an amazing amount of control and flexibility to create complex web applications. For further information about some of these functions see the API chapter later in the manual as well.

### Allow public access to a page

To allow public access to a page we make use of the special usergroup called PUBLIC as follows:-

```
<?php
$groupswithaccess="PUBLIC";
require_once("slpw/sitelokpw.php");
?>
```

Within the page you can then use the `$slpublicaccess` variable to test if the current user is a public visitor (set to `true`) or a logged in member (set to `false`).

If you specify any groups as well as PUBLIC then members are handled slightly differently. For example

```
<?php
$groupswithaccess="PUBLIC,GOLD";
require_once("slpw/sitelokpw.php");
?>
```

would ensure that only members of GOLD get full access. All other users, even if they are logged in, would be treated as a public user.

Below are some code examples for use on pages that control who see what on a page.

## Display section of page only for public visitors

You can place a special PHP tag at the start and end of a page section (any html or javascript etc) so that it will only be displayed to public visitors as follows:-

```
<?php if ($slpublicaccess) { ?>  
html code to be displayed only for public users  
<?php } ?>
```

For example you could have a login form displayed only when a user is not logged in and remove it for users already logged on (displaying a logout link instead perhaps).

## Display section of page only for members

You can place a special PHP tag at the start and end of a page section (any html or javascript etc) so that it will only be displayed to logged in members as follows:-

```
<?php if (!$slpublicaccess) { ?>  
html code to be displayed only for logged in members  
<?php } ?>
```

## Display section of page for members of specific group

Here is some example code showing how to display as section only for members of a certain group (in this case GOLD) and even if they are expired.

```
<?php if (sl_ismemberof("GOLD")){ ?>  
html code to be displayed only for logged in members of GOLD  
<?php } ?>
```

Notice this uses the Sitelok function [sl\\_ismemberof\(\)](#) which returns true if the user is a member of the group (even if expired) and false if not.

## Display section of page for active members of specific group

Here is some example code showing how to display a section only for members of a certain group that has not expired (in this case GOLD) and even if they are expired.

```
<?php if (sl_isactivememberof("GOLD")){ ?>
html code to be displayed only for logged in active members of GOLD
<?php } ?>
```

Notice this uses the Sitelok function [sl\\_isactivememberof\(\)](#) which returns true if the user is a member of the group (and not expired) and false if not.

## Display section of page for specific user

Here is some example code showing how to display a section only for a specific user.

```
<?php if ($slusername=="username"){ ?>
html code to be displayed only for username
<?php } ?>
```

## Login link and redirect back to this page

You can add a Login link or button to PUBLIC pages so that users can login and be returned to the same page as logged in members. To do this you can point the login link to `/slpw/login.php` with a redirect override like this

```
/slpw/login.php?redirect=<?php echo $startpage; ?>
```

## Download links for public visitors

The [sitelinklink\(\)](#) function can be used on PUBLIC pages as well as members only pages. For further details about using that function see chapter 5 (Securing download links).

If you wish to have a download link on a PUBLIC page but require the user to login before access is allowed then this can be done with the function

```
sitelinklogin("test.zip",1)
```

This works in the same way as the standard [siteloklink\(\)](#) function except that if the user is not currently logged in it will display the login template before access is allowed. It has two ways of working.

### Method 1 - Login and return to link

In this mode if the user is not logged in they will be asked to login and then get returned to originating page with the link so they can click it to download. If the page is long or contains a lot of links you can use anchors containing the filename in which case the user will be returned exactly to the that link. For example

```
<a href="<?php siteloklinklogin('test.zip',1)?>"
name="test.zip">Download here</a>
```

### Method 2 - Download landing page

In this mode if the user is not logged in they will be asked to login and then taken to a download landing page which will start the download after a few seconds and contains a return link back to the originating page. The download landing page should be in the slpw folder and is enabled like this.

```
<a href="<?php siteloklinklogin('test.zip',"downloadpage.php")?
">>Download here</a>
```

downloadpage.php is an example included in the slpw folder which you can use and customise as required. Within the page you can use the following placeholder variables.

**!!!link!!!** This inserts a link to the download which can be clicked by the user if the automatic download does not start for some reason.

**!!!returnpage!!!** This inserts the URL of the originating page.

Here is a simple example landing page.

```
<html>
<head>
<title></title>
</head>
<body>
Your download should begin soon but if not please click the link
below <br>
<br>
<a href="!!!link!!!">!!!filename!!!</a><br>
<br>
```

```
Click <a href="!!!returnpage!!!">here</a> to return.  
</body>  
</html>
```

### Allow user to join a new usergroup

Sitelok can generate a link that when clicked will add the user to the specified usergroup. The following will create the link

```
<?php  
sitelokaddgroup("GROUP", "60", "email.htm", "admin.htm", "thanks.php"); ?  
>
```

"GROUP"	The group name to add.
"30"	The expiry date. example values
"30"	Sets the expiry date 30 days from now
"161209"	sets the expiry date to 16th December 2009 assuming DDMMYY format is set
"email.htm"	The email template sent to the client when updated. Set to "" if not required.
"admin.htm":	The email template sent to the admin when updated. Set to "" if not required.
"thanks.php"	URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php  
sitelokaddgroup("GOLD", "365", "email.htm", "admin.htm", "thanks.php"); ?  
>">Join GOLD area for 1 year</a>
```

See the email chapter for details about similar links for use in email templates.

### Allow user to remove themselves from a usergroup

Sitelok can generate a link that when clicked will remove their membership from the specified usergroup. The following will create the link

```
<?php  
sitelokremovegroup("GROUP", "email.htm", "admin.htm", "thanks.php"); ?>
```

"GROUP"	The group name to remove.
"email.htm"	The email template sent to the client when updated. Set to "" if not required.

- "admin.htm":** The email template sent to the admin when updated. Set to "" if not required.
- "thanks.php"** URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php  
sitelokremovegroup("GOLD","email.htm","admin.htm","thanks.php");?  
>">Unsubscribe from GOLD area</a>
```

See the email chapter for details about similar links for use in email templates.

### **Allow user to replace a usergroup with a new one**

Sitelok can generate a link that when clicked will replace the users membership of one group with another and optionally extend that membership. The following will create the link

```
<?php  
sitelokreplacegroup("GROUP","NEWGROUP","30","email.htm","admin.htm","  
thanks.php"); ?>
```

- "GROUP"** The existing group name to replace.
- "NEWGROUP"** The new usergroup
- "30"** The expiry date. example values
  - "" Leave expiry date as it is
  - "30" Sets the expiry date 30 days from now
  - "+30" Extends any existing expiry by 30 days
  - "161209" sets the expiry date to 16th December 2009 assuming DDMMYY format is set
- "email.htm"** The email template sent to the client when updated. Set to "" if not required.
- "admin.htm":** The email template sent to the admin when updated. Set to "" if not required.
- "thanks.php"** URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php  
sitelokreplacegroup("SILVER","GOLD","", "email.htm","admin.htm","thank  
s.php");?>">Upgrade from Silver to Gold membership</a>
```

See the email chapter for details about similar links for use in email templates.

## Allow user to extend a group membership

Sitelok can generate a link that when clicked will extend the users membership of a group. The following will create the link

```
<?php  
sitelokextendgroup("GROUP","30","0","email.htm","admin.htm","thanks.p  
hp"); ?>
```

"GROUP"	The group to be extended
"30"	The expiry for the group. Here are some examples
"365"	This would set the new group with an expiry 365 days from now
"010809"	This would set the expiry to 1st August 2009 assuming DDMMYY format
"+30"	This would extend the current expiry by 30 days
"0"	This setting can be used to determine whether to extend the expiry. Here are some examples
"0"	Setting 0 will always adjust the expiry date
"+30"	Using a + in front of the number will only adjust the expiry date if the existing expiry was 30 (in this case) or more days ago.
"-30"	Using a - in front of the number will only adjust the expiry if there are at least 30 (in this case) days until the current expiry.
"email.htm"	The email template sent to the client when updated. Set to "" if not required.
"admin.htm":	The email template sent to the admin when updated. Set to "" if not required.
"thanks.php"	URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php  
sitelokextendgroup("GOLD","30","0","email.htm","admin.htm","thanks.ph  
p");?>">Extend Gold membership for 30 days</a>
```

See the email chapter for details about similar links for use in email templates.

## Allow user to delete their account

Sitelok can generate a link that when clicked will delete the users account.

```
<?php sitelokdelete("email.htm","admin.htm","thanks.php") ?>
```

- "email.htm"** The email template sent to the client when updated. Set to "" if not required.
- "admin.htm":** The email template sent to the admin when updated. Set to "" if not required.
- "thanks.php"** URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php sitelokdelete("email.htm","admin.htm","goodbye.php");?>">Close your account</a>
```

As this is a non reversible action we advise that you first use a hyperlink pointing to a 'are you sure page' which contains the above link.

## Send email to user

Sitelok can generate a link that when clicked will send a defined email template to the user (and to the admin if required).

```
<?php siteloksendemail("email.htm","admin.htm","thanks.php") ?>
```

- "email.htm"** The email template to send to the user. Set to "" if not required.
- "admin.htm":** The email template to send to the admin. Set to "" if not required.
- "thanks.php"** URL of the page to redirect to.

In practice you would use this link as part of a hyperlink or button. e.g.

```
<a href="<?php siteloksendemail("email.htm","admin.htm","goodbye.php");?>">Send reset password email</a>
```

See the email chapter for details about similar links for use in email templates.

## Chapter 14 Search engine access

Normally you will want your members area pages to be secure and visible only to members. However for some sites it may be desirable to still allow search engines to have access to these pages so that they can be indexed.

Please be aware that providing content to a search engine that is different to what is viewed by a normal public visitor may be in violation of the search engines policy.

### Enable search engines to index pages

To enable search engine access select **Tools - Configuration - General settings** and set **Allow search engine access** to **On**. You must also enter the user group that the search engine will be seen as. This enables you to have areas indexed and not others. If you want all pages indexed then set this group to **ALL** (which is the default). When search engines visit they will not login as such (most do not support sessions or cookies anyway) but will have access to the page content. Remember though that they may not see your site exactly as your members do.

We strongly recommend adding the following meta tag to the HEAD section of your pages to stop search engines keeping a cached copy of your pages which are visible to anyone. This way the pages are only indexed as required.

```
<meta name="robots" content="noarchive">
```

You can also set your sites robots.txt file to control search engines as well.

### Overriding the setting for individual pages

You can also override the global setting made in the configuration page on a page by page basis. To do this add the following setting to the code snippet at the top of the page before the call to `sitelokpw.php`:-

```
$allowsearchengine=true;  
$searchenginegroup="ALL";
```

or to disable access for a certain page:-

```
$allowsearchengine=false;
```

## The search engine user

Search engine robots don't normally support sessions or cookies etc so they will not login to Sitelok in the normal way. Instead we let them have access to your pages directly if the usergroup set would normally have access. While visiting the page the `$slusername` will be set to `searchenginebot` so you can test for this programmatically if required. You could also use this username for pages that are controlled by username rather than usergroup. The following two variables will also be set.

`$slsearchengine` Set to true if search engine access (false otherwise)  
`$slsearchenginebot` Set to the accessing robot name. Currently these are

`googlebot`  
`slurp`  
`msnbot`  
`ia_archiver`  
`charlotte`  
`teoma`  
`speedy`

## Verifying search engines are who they say

Allowing search engine access is not quite as easy as it sounds because we need to be sure that the search engine robot is real and not just a hacker pretending to be a search engine robot. Some systems simply check the user agent of the visitor which although will work is extremely easy to fake and therefore is not good enough for Sitelok. Instead we check that the host name of the IP address used matches with the host that is used by the search engine. We do this using a forward and reverse lookup to make cheating much harder. So that we don't have to do this for every visitor we first check the user agent for supported search engines and if matched continue with the search engine verification.

Having said all that (if you are still here!) we should point out that allowing search engine access is always going to introduce some weaknesses. For this reason we provide access to only the following reputable search engines currently.

[Google](#)  
[Yahoo](#)  
[Inktomi](#)

[MSN](#)

[Alexa \(Amazon\)](#)

[SearchMe](#)

[Ask Jeeves](#)

[Speedy \(Entireweb\)](#)

We may add to this list upon request. Remember that search engines such as google allow searchers to see cached copies of pages seen by the search engine robot so we recommend you add the following meta tag to your members pages to stop caching. This way the pages will be indexed only.

```
<meta name="robots" content="noarchive">
```

You can also set your sites robots.txt file to control search engines as well.

# Chapter 15 Configuration

When you install Sitelok the default settings are usually fine for most membership areas. However Sitelok provides many extra features and settings that you can set. Most of these can be set in the configuration page [Tools - Configuration](#).

We will go through each of the settings available in this chapter. You may not need to use most of these settings but a glance through this chapter may be useful to give you an idea of the things you can do if required.

## General Settings

### Site name

This is the name of your membership area and is displayed by the template variable `!!!sitename!!!`.

### Admin email address

This is the main email address used by the administrator of the members area. You can also define a secondary email address.

### Date format

This is the date format used by Sitelok. Once set during installation you should not change this again. You can choose either `DDMMYY` or `MMDDYY`.

### Site Key

This is a random key string used by some parts of Sitelok such as download links. This should not be changed after you start using your site and you should keep a copy of the value used.

### Maximum session time

Sitelok by default allows sessions of unlimited time (although your server may limit sessions internally). You can if you wish set the required maximum session time in seconds. For example 3600 would allow users to

be logged in only for one hours. After this any page access would require them to login again. Set to 0 for unlimited sessions.

## Maximum inactivity time

You can set the maximum time in seconds a user can be logged in with no activity (clicking links to secured pages). By default this is unlimited (set to 0) but you could set this to any time required. For example enter 600 for 10 minutes.

## Login CAPTCHA

This setting enables or disables the requirement for a turing code (CAPTCHA) to be entered during login. See the section about login pages and templates to see how to display and collect the turing code. If you are using the built in default login template then enabling this option will display and use the turing code without any other settings required.

## Register CAPTCHA

This setting enables or disables the requirement for a turing code (CAPTCHA) to be entered during user registration. See the section about registration pages to see how to display and collect the turing code.

## Remember me (cookie login)

This setting determines whether Sitelok attempts to store a users username and password in a cookie so that they are either displayed in the login form for them or automatically log the user in with no login form being displayed. See the section about login pages and templates for more information about this settings. You can choose between the following settings.

[Disabled](#)  
[Remember Me](#)  
[Auto Login](#)

## Allow concurrent logins

As with most membership systems Sitelok will allow a username and password to be used whenever they are needed. This means that several people could share a username and be logged in at the same time (concurrently). For most sites this is not a problem but you can block

concurrent access using this setting if you wish. In this case if someone logs in then any existing user already logged in with the same username will have their session cancelled.

### **Force database read on page access**

Normally once a session has begun the users details are stored in session variables for efficiency because it saves making unnecessary calls to the mysql database. However if you were to update some of the user data (for example a custom field value) from the admin panel or via the Sitelok API then this value wouldn't be updated for display on the members pages until the user starts a new session. In most cases this is not important but you can force Sitelok to either always read this data from the database or to do this on selected pages.

You can force Sitelok to read the user data from the database on each page access by enabling this setting.

To simply enable this on a page by page basis just add this setting to the page before the call to sitelokpw.php.

```
$dbupdate=true;
```

### **Allow search engine access to protected pages**

If set to Yes Sitelok will allow supported search engine robots to index your protected pages. You will need to set the user group that the search engine will belong to as well. See the chapter about search engine access for more details.

### **Search engine usergroup**

If you allow search engines access to members pages then you need to tell Sitelok which user group they should belong to. By default this is set to ALL. You can enter multiple groups by separating them with commas. Don't enter the group ADMIN as this is blocked anyway for obvious reasons!

### **Email change verification**

If enabled (set to Yes) then whenever a user changes their email address in modify profile page they will be sent an email to the requested address to confirm this. Only after the link in the email is clicked will the email address

be updated. You can set the email template to send out in this case. You can use the following email template variables in the template.

**!!!verifyemail!!!**

This inserts a link that when clicked will update the email address. This should normally be used in a hyperlink like this

```
<a href="!!!verifyemail!!!">click to confirm</a>
```

**!!!requestedemail!!!**

The requested email.

If you prefer to have the email confirmation link sent to the existing email address instead of the requested email address you should set the template to 'No Email' and use the above template variables in the user email template defined for the form.

### **Email field must be unique**

By default Sitelok only requires that the username field is unique. However you can if required ensure that an email address is only used by one user. You can do this only for user entry (registration and modify profile forms) or for completely (for admin entry and API calls as well). If you enable this feature you may consider enabling 'Email change verification' as well.

### **Login with username or email**

If set to Yes then users can login using their email address in place of the username. If set to No then only the username (and password) can be used (unless the username is the email as well of course).

## **Passwords**

### **Forgotten password email template**

This setting allows you to select the email template sent to users when they request a forgotten password from the login form. The drop down menu will display all of the available email templates that are stored in the slpw/email folder. You can create new templates and edit existing ones using Sitelok's built in html editor as described earlier. Remember that if you enable hashed passwords that you should use a template that does not try to

display the existing password but instead provides the user with a new one which they can activate (such as the example `forgotpasshash.htm`).

## Random password mask

There are several areas within Sitelok where random passwords are generated such as the Add user form in the control panel. You can define the format of the random passwords generated using a string of characters. Each character in the mask defines which type of character will appear in the password in that position.

**c** will be a lowercase character a-z

**C** will be an uppercase character A-Z

**X** will be a lower or uppercase character

**#** will be a digit 0-9

**A** will be any character a-z, A-Z or 0-9

**U** will be any character from the list of allowed characters

The default setting of `cccc##` will generate 4 lower case characters followed by 2 digits.

You can also override the default list of characters valid in passwords by adding this setting to `slconfig.php`

```
$ValidPasswordChars="abcdefghijklmnopqrstuvwxyz";
```

## Profile update password required

This setting determines whether a user is required to enter their existing password in an update profile form before changes are made. You can set if the password is required for any changes, just for updating the password or not at all.

You can override this setting for specific profile forms by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$profilepassrequired=0;
```

Set to 0 if no password is required, 1 for if required for any changes and 2 only for password change.

## Store password as hashes

By default Sitelok stores members passwords in mysql using plain text enabling forgotten passwords to be retrieved directly. However its recommended to enable hashed passwords to keep your users passwords secure. This means that if a hacker were to get access to MySQL on your server he will not see your users passwords which may of course be the same password they use on other sites.

Once enabled Sitelok will only know the actual password when a user initially registers after which only the hash is known, This means that Sitelok cannot email the users password if they forget it. Instead if a user forgets their password and uses the forgotten password option on login they will receive an email with a new random password which will be activated by clicking a link. They can of course change this password using the modify profile page once they are logged in, if you allow this.

**If you enable this feature Sitelok will convert your existing users passwords to secure hashes when the user next logs in. Passwords that are hashed can not be reversed.**

As mentioned there is no point to display passwords in email templates when storing hashed password so instead of using `!!!password!!!` you can use `!!!passwordifknown!!!` which will display the password if it has just been created (during registration for example) and once hashed will display `*****` instead. You can change the `*****` to any other text by setting

```
define("MSG_PASSWORDHASHED", "*****");
```

in `slconfig.php` as needed.

If you wish to allow users to retrieve a forgotten password you should update the email set to display a new password using `!!!newpassword!!!` and allow the user to activate it using a `!!!activatepassword!!!` link. You can use the example email template named `forgotpasshash.htm`. See the email section of the manual for more details.

## Pages & Templates

### Logout page URL

This is the default page that users are sent to when they click a logout link. It should be a full URL such as <http://www.yoursite.com/logout.php>. You can override this setting by adding `&page=/folder/page.php` to your logout links.

```
/slpw/sitelokpw.php?sitelokaction=logout&page=/folder/page.php
```

### Message and error template

By default when a message or error needs to be displayed to a user it is shown as a text message on a blank page. You can use your own page to display messages though by setting the URL in this field. For example

```
http://www.yoursite.com/messages.php
```

Insert the following code in the page where you wish to display the message.

```
<?php
echo htmlentities(urldecode($_GET['slmessage']), ENT_QUOTES, 'UTF-8');
?>
```

If you need to you can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$messagepage="http://www.yoursite.com/messages.php";
```

### Login template

This is the login form displayed by Sitelok when a user visits a secured page and they are not already logged in. By default Sitelok has a built in form which can be selected by leaving this setting blank. If you wish to replace the login form with your own template page then create the page as described in the login template chapter, upload it to the `slpw` folder and set the name in this setting (no path required). For example.

```
logintemplate.php
```

Please note that the login template is different to a login page and is never directly accessed by a user.

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$loginpage="logintemplate.php";
```

### Expired membership page URL

By default when a users membership to a particular user group has expired the message page is displayed with the text

[Access to this page is blocked because your membership has expired](#)

However you can create a custom page for this purpose that could perhaps contain a buy button enabling them to renew membership etc. Enter the full URL to the required page such as

```
http://www.yoursite.com/expired.php
```

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$expiredpage="http://www.yoursite.com/expired.php";
```

You can also allow access to certain secured pages even when the users membership has expired by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$allowexpireduser="Y";
```

This is useful on pages where you still want to ensure the users has logged in but the fact that their membership has expired is not important. A good example is on an `$expired` page itself.

### Wrong group page

Normally when a user is logged in but tries to access a page that is accessible only to other user groups the message page displays the following text

[Your membership does not allow access to this page](#)

You can create a custom page for this purpose if you prefer. Enter the full URL to the required page such as

<http://www.yoursite.com/wronggroup.php>

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$wronggroupage="http://www.yoursite.com/wronggroup.php";
```

### No access page URL

Normally when a user tries to access a secured page and they are not logged in Sitelok will display the login form. On some sites you may prefer users not to bookmark and access pages in this way. Instead you can have users redirected to the page specified in this setting. This page could be your login form perhaps or alternatively a page explaining how to join the members area. Enter the full URL to the page. For example

<http://www.yoursite.com/noaccesspage.php>

You can override this setting on individual secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$noaccesspage="http://www.yoursite.com/noaccesspage.php";
```

If you have this setting pointing to a login page you can optionally set a login redirect flag as well. To have the user redirect back to the page they came from add `?redirect=page` to the URL. To redirect to the usergroup start page then add `?redirect=user`. For example

<http://www.yoursite.com/noaccesspage.php?redirect=page>

## Log Settings

### Log entry types

You can select which kind of entries will get stored in the log. By default all entry types are logged. You can select from

- [Login / Logout](#)
- [Login Problems](#)
- [Password Requested](#)

- [Download](#)
- [Download problems](#)
- [Email sent](#)
- [Membership expired \(where user attempted to access\)](#)
- [User modified details](#)
- [API function call \(includes user registration\)](#)

### Server path to text format log file

This version of Sitelok uses a mysql table to store the access log which provides much greater control. Earlier version used a text file for the log. If you wish to also continue to have a text log file you can set the full server file path to the file here. For example

[/home/domain/public\\_html/df64gvtyerfgdt/accesslog.txt](/home/domain/public_html/df64gvtyerfgdt/accesslog.txt)

### Custom fields

#### Custom X field title

This is the title that is displayed in the admin panel for the custom field. If you leave the title blank then the custom field will not be displayed in the admin. Please note that this has no connection with the name used in mysql table which must not be changed.

#### Custom X size

By default each custom field stores up to 255 characters. You can optionally increase the maximum size to [65535](#) or [16777215](#) characters. Only do this for the fields that require this. Once a field is set to a larger size you will see it displayed as a text area in the add user and edit user pages. Also in the form designers the text area field type will be available for that field.

#### Custom X Validation

The validation settings can usually be left as [No Validation](#). The form designers now have validation options that can be used anyway. If you do enable validation here then the field will be required and custom validation code can be added to eventhandler.php.

If validation is enabled then Sitelok will ensure a value is entered for that custom field. You can set if the value is required for user entry (registration

or modify profile forms), admin entry or both. If you wish Sitelok to do further validation apart from checking a value is entered then you can modify the validation functions for each custom field in the file `eventhandler.php` (in the `slpw` folder). If you update `eventhandler.php` then make sure you keep a copy of your modifications when you upgrade Sitelok.

## Dashboard

### Main table column order

Here you can set which fields (columns) are displayed in the main dashboard table and also in which order. To do this check the fields you wish to display and drag and drop to the required order. Notice that the Select and the Action columns must be displayed.

### Default rows to display in admin page

This selects the default number of user records to display per page in the control panel.

### Main table font size

This selects the the font size of the main user table on the dashboard.

## User page layout

In this section you can change the order custom fields are displayed on the add user and edit user pages using drag and drop. You can also move custom fields to the User Details box if you prefer.

## Download paths

### Full file path to download folder

This is the full server file path to the folder where you keep download files that you create secure links for. During installation a folder with a random name (starting with `sl_files` as of V4.0) is created for this purpose but you can change this setting here. You can also create subfolders in this location. An example is

`/home/domain/public_html/slfiles_df64gvtyerfgdt/`

The last / character is required. On some servers you may be able to set this outside of the web root for more security. However using a random folder name is fine as it cannot be found by search engines or robots as long as you do not directly link to any files in the folder.

Although this folder is secured by its random name we also attempt to place a .htaccess file in the folder when it is created containing the following.

```
order allow,deny
deny from *
```

which should stop direct access even if the folder name is known (on servers that support htaccess).

You can override this setting on secure pages by adding this line to the code snippet at the top before the call to `sitelokpw.php`.

```
$filelocation="/home/domain/public_html/slfiles_df64gvtyerfgdt/";
```

### Additional download folders

Normally the default download folder (and subfolders) is sufficient but you can add as many additional folders as required by clicking the [Add location button](#). Give each a unique name. Sitelok also supports Amazon S3 as a file location. See the section about securing download links for more information. If you need to create more folders than there are input fields for simply save the configuration and open again the page to get more empty fields.

### Email settings

#### Send email using

By default Sitelok sends email using the PHP `mail()` function. Sitelok does also support the user of the PHPmailer library as well. PHPmailer sends email using an smtp server in a similar way to an email client. If you select this option then you need to also enter the following settings.

- Email account username
- Email account password
- Email account smtp server
- Email account port

[Email account server authentication \(enabled/disabled\)](#)  
[Email account security \(No SSL/SSL/TLS\)](#)

## Reply to email override

By default the reply to email will be the email set in the [Admin Email](#) setting but this can be overridden with this setting.

## Allow from name and reply-to override on send email dialog

If you enable this option then when sending emails from the control panel you will have the option to override the default from name and reply-to email address.

## Send copy of emails sent from dashboard

If you wish to be sent a copy of all emails sent from the control panel using send email, send email to use or send email to selected users you can enable this option. Enter the address to receive the copies. The subject line of the email will contain the [\(username\)](#) appended to it so you know who it was sent to. When sending bulk emails only one copy will be sent and [\(X selected users\)](#) will be appended to the email subject.

You can set copies to be always sent or have a checkbox on the send email dialog to choose if a copy is sent.

## Delay between emails during bulk delivery

Some hosting companies limit the speed at which you can send multiple emails using their servers. This settings allows you to set a short delay between each email sent.

## Send test email

Click this button to send a test message to the Admin Email address.

## Install paths

### Full file path to the Sitelok folder

This should be set to the full server file path to the folder where you installed Sitelok. The installation script normally determines this path for you but if not you can set it here. An example is

[/home/domain/public\\_html/slpw/](#)

The last / character is required.

### URL to Sitelok folder

This is the full URL to the folder where you installed Sitelok. The installation script normally determines this URL for you but if not you can set it here. An example is

<http://www.yoursite.com/slpw/>

The last / character is required.

### Full file path to email template folder

This should be set to the full server file path to the folder where you store email templates. The installation script normally determines this path for you but if not you can set it here. An example is

[/home/domain/public\\_html/slpw/email/](#)

The last / character is required.

### URL to email template folder

This is the full URL to the folder where you store email templates. The installation script normally determines this URL for you but if not you can set it here. An example is

<http://www.yoursite.com/slpw/email/>

The last / character is required.

## Full file path to backup folder

This should be set to the full server file path to the folder where you store backup files. During installation a folder with a random name (starting with `slbackups_`) is created in the root but you can change this if required.

`/home/domain/public_html/slbackups_er6r25e5asfw/`

The last `/` character is required.

Although this folder is secured by its random name we also attempt to place a `.htaccess` file in the folder when it is created containing the following.

```
order allow,deny
deny from *
```

which should stop direct access even if the folder name is known (on servers that support `htaccess`).

We do advise to download and delete backup files as soon as they are created for maximum security.

## Chapter 16 Sitelok API

This chapter is for PHP developers who need to integrate Sitelok with other systems or enhance their members area with PHP scripting. Most users won't need anything here but may like to read through to have an idea of what can be done.

For more advanced applications Sitelok provides an API that allows your PHP applications direct access to add and modify users. To use the API simply include `sitelokapi.php` in your application (either in a protected page or normal page) as follows:-

```
require_once"slpw/sitelokapi.php";
```

There are in general two types of API functions. Functions with the `slapi_` prefix are designed to work on the specified user and can be used from external applications. Functions with the `sl_` prefix are designed to work on the current logged in user so should only be called from members pages and not external applications. The `sl_` function calls will update the Sitelok variables stored in the current session as well.

### **slapi\_getuser**

```
$result=slapi_getuser($user,$created,$pass,$enabled,$name,$email,  
$groups, $custom1,$custom2,.....$custom50);
```

This function will retrieve the information for a specific user. `$user` must be set to the required username. After the call the function returns -1 if there was a database error, 0 if the user was not found or 1 if found. The users details are returned in the other parameters. The `$created` parameter contains the date the user was originally created and is always in the 6 character format `YYMMDD`.

### **slapi\_adduser**

```
$result=slapi_adduser($user,$pass,$enabled,$name,$email,$groups,  
$clientemail,$adminemail,$logit,$custom1,$custom2,  
$custom3,.....,$custom50);
```

This function will add a new user to the Sitelok database. After the call the function returns -1 if there was a database error, 0 if the user was not added (usually because the user already exists) or 1 if added. Most of the parameters are self explanatory but a few need further details.

<b>\$enabled</b>	Should be set to either "Yes" or "No".
<b>\$clientemail</b>	Filename of the email template to send to the client if added successfully. For no email set to "".
<b>\$adminemail</b>	Filename of the email template to send to the site Administrator if added successfully. For no email set to "".
<b>\$logit</b>	Set to 1 to have the API call logged in the Sitelok log. Set to 0 to ignore.
<b>\$custom</b>	The custom fields are optional and you only need to set values for the fields that are actually being used on your site.

## slapi\_modifyuser

```
$result=slapi_modifyuser($user,$pass,$enabled,$name,$email,$groups,  
    $clientemail,$adminemail,$logit,  
    $custom1,$custom2,$custom3,.....,$custom50);
```

The `slapi_modifyuser` function allows you to modify the details of an existing user. After the call the function returns -1 if there was a database error, 0 if the user was not modified (usually because the user does not exist) or 1 if modified. Most of the parameters are self explanatory but a few need further details.

<b>\$enabled</b>	Should be set to either "Yes" or "No".
<b>\$clientemail</b>	Filename of the email template to send to the client if added successfully. For no email set to "".
<b>\$adminemail</b>	Filename of the email template to send to the site Administrator if added successfully. For no email set to "".
<b>\$logit</b>	Set to 1 to have the API call logged in the Sitelok log file. Set to 0 to ignore.
<b>\$custom</b>	The custom fields are optional and you only need to set values for the fields that are actually being used on your site.

## slapi\_deleteuser

```
$result=slapi_deleteuser($user,$clientemail,$adminemail,$logit);
```

This function deletes a user from the Sitelok database (including log entries). After the call the function returns -1 if there was a database error, 0 if the user was not deleted (usually because the user does not exist) or 1 if deleted.

<b>\$user</b>	Username of the user to delete
<b>\$clientemail</b>	Filename of the email template to send to the client if added successfully. For no email set to "".
<b>\$adminemail</b>	Filename of the email template to send to the site Administrator if added successfully. For no email set to "".
<b>\$logit</b>	Set to 1 to have the API call logged in the Sitelok log file. Set to 0 to ignore.

## sl\_delete

```
$result=sl_deleter($clientemail,$adminemail);
```

This function deletes the current user completely. After the call the function returns -1 if there was a database error, 0 if the user was not deleted or 1 if deleted. This function can only be called from a members page for use on the current user.

<b>\$clientemail</b>	Filename of the email template to send to the client if added successfully. For no email set to "".
<b>\$adminemail</b>	Filename of the email template to send to the site Administrator if added successfully. For no email set to "".

## slapi\_addorupdate

```
$result=slapi_addorupdate($user,$pass,$email,$name,$group,  
    $groupexpiry,$clientemail,$adminemail,$clientemailm,  
    $adminemailm, $logit,  
    $custom1,$custom2,$custom3,.....,$custom50);
```

The `slapi_addorupdate` function is probably the most useful of the API functions as it will add a new user if the user does not exist already or update an existing user. In addition, unlike the other functions that work on the raw data from the database, `slapi_addorupdate` allows you to define multiple user groups each with their own expiry times defined in days. If a user already exists this function will automatically adjust the expiry dates of groups the user is already a member of and add new ones as required.

It should be noted that if a user already exists then only the groups and expiry times are updated. The values of other fields are left as is in case a user has intentionally updated them manually.

After the call the function returns -1 if there was a database error, 0 if the user was not added or modified or 1 if added or modified successfully. Most of the parameters are self explanatory but a few need further details.

<b>\$pass</b>	If set then this password will be used if a new user is created. If left blank then Sitelok will generate a random password consisting of 4 alphanumeric characters followed by 2 numeric digits (e.g. hety53).
<b>\$groups</b>	This is an array of the groups you wish to add or modify for the user. Each array element should contain only the group name with no expiry date.
<b>\$groupexpiry</b>	This is an array with each element containing the number of days of membership allowed for each of the groups defined in \$groups. Just set the number of days and the function will automatically adjust existing expiry times or create new ones. Set to 0 for no expiry time.
<b>\$clientemaila</b>	Filename of the email template to send to the client if added as a new user. For no email set to "".
<b>\$adminemaila</b>	Filename of the email template to send to the site administrator if added as a new user. For no email set to "".
<b>\$clientemailm</b>	Filename of the email template to send to the client if modified as an existing user. For no email set to "".
<b>\$adminemailm</b>	Filename of the email template to send to the site administrator if modified as an existing user. For no email set to "".
<b>\$logit</b>	Set to 1 to have the API call logged in the Sitelok log file. Set to 0 to ignore.
<b>\$custom</b>	The custom fields are optional and you only need to set values for the fields that are actually being used on your site.

Here is an example call for `slapi_addormodify`.

```
$groups[0]="GROUP1";
$groups[1]="GROUP2";
$groupexpiry[0]=30;
$groupexpiry[1]=365;
slapi_addorupdate("username","12345","user@site.com",
"Adrian Jones",$groups,$groupexpiry,"newuser.htm",
"newuseradmin.htm","updateuser.htm","updateuseradmin.htm",1);
```

## slapi\_getallusernames

```
$users=slapi_getallusernames($orderby,$where);
```

The `slapi_getallusernames` function returns an array of strings containing the username for all users in the database. If a database error occurred then -1 is returned.

The `$orderby` and `$where` parameters are optional.

`$orderby` Set to the field name to sort the results by such as Username.

`$where` You can set a Mysql where statement to filter results such as Usergroups like '%ADMIN%'

For example the following will return an array of all usernames that are members of the ADMIN group. The list will be ordered by the Name field in descending order.

```
$users=slapi_getallusernames("Name DESC","Usergroups like '%ADMIN%'");
```

## slapi\_getallusergroups

```
$groups=slapi_getallusergroups();
```

The `slapi_getallusergroups` function returns an array containing the usergroups defined in Sitelok admin. The array contains the name, description, logon action and login value as follows:

`$groups['name']` Contains the group name such as ADMIN  
`$groups['description']` Contains the group description  
`$groups['loginaction']` Contains the login action. This could be one of the following:-

`URL` This means this group redirects to a fixed URL upon login.

`custom1` This means this group redirects to a user specific URL set in the `custom1` field. `custom1` to `custom50` are supported.

`$groups['loginvalue']` Contains the URL to redirect to.

## slapi\_addgroup

```
$result=slapi_addgroup($username,$group,$groupexpiry,  
    $clientemail,$adminemail);
```

The `slapi_addgroup` function adds the user to a certain usergroup.

<code>\$username</code>	The user to update
<code>\$group</code>	The group name to add.
<code>\$groupexpiry</code>	The expiry date. example values "30" Sets the expiry date 30 days from now "161209" sets the expiry date to 16th December 2009 assuming DDMMYY format is set
<code>\$clientemail</code>	The email template sent to the client when updated. Set to "" if not required.
<code>\$adminemail</code>	The email template sent to the admin when updated. Set to "" if not required.

After the call the function returns -1 if there was a database error, 0 if the user was not updated (usually because the user does not exist or already is a member of the group) or 1 if updated successfully.

## sl\_addgroup

```
$result=sl_addgroup($group,$groupexpiry,$clientemail,$adminemail);
```

The `sl_addgroup` function adds the currently logged in user to a certain usergroup. This is the same as the `slapi_addgroup` function except that it works from a members page on the currently logged in user. The Sitelok variables and current session are updated as well.

## slapi\_removegroup

```
$result=slapi_removegroup($username,$group,$clientemail,$adminemail);
```

The `slapi_removegroup` function removes the user from a certain usergroup.

<code>\$username</code>	The user to update
<code>\$group</code>	The group name to remove.

- \$clientemail** The email template sent to the client when updated. Set to "" if not required.
- \$adminemail** The email template sent to the admin when updated. Set to "" if not required.

After the call the function returns -1 if there was a database error, 0 if the user was not updated (usually because the user or the group does not exist) or 1 if updated successfully.

## sl\_removegroup

```
$result=sl_removegroup($group,$clientemail,$adminemail);
```

The `sl_removegroup` function removes the currently logged in user from a certain usergroup. This is the same as the `slapi_removegroup` function except that it works from a members page on the currently logged in user. The Sitelok variables and current session are updated as well.

## slapi\_replacegroup

```
$result=slapi_replacegroup($username,$group,$newgroup,$groupexpiry,  
    $clientemail,$adminemail);
```

The `slapi_replacegroup` function replaces a users membership of one group with another and allows the expiry to be adjusted at the same time.

- \$username** The user to update
- \$group** The existing group name to replace.
- \$newgroup** The new group to replace the existing one
- \$groupexpiry** The expiry date. example values
  - "" Leave expiry date as it is
  - "30" Sets the expiry date 30 days from now
  - "+30" Extends any existing expiry by 30 days
  - "161209" sets the expiry date to 16th December 2009 assuming DDMMYY format is set
- \$clientemail** The email template sent to the client when updated. Set to "" if not required.
- \$adminemail** The email template sent to the admin when updated. Set to "" if not required.

After the call the function returns -1 if there was a database error, 0 if the user was not updated (usually because the user does not exist or is not a member of the group) or 1 if updated successfully.

## sl\_replacegroup

```
$result=sl_replacegroup($group,$newgroup,$groupexpiry,  
    $clientemail,$adminemail);
```

The `sl_replacegroup` function replaces the currently logged in users membership of one group with another and allows the expiry to be adjusted at the same time. This is the same as the `slapi_replacegroup` function except that it works from a members page on the currently logged in user. The Sitelok variables and current session are updated as well.

## slapi\_extendgroup

```
$result=slapi_extendgroup($username,$group,$groupexpiry,$expirytype,  
    $clientemail,$adminemail);
```

The `slapi_extendgroup` function extends the expiry date of a group for a certain user.

<code>\$username</code>	The user to update
<code>\$group</code>	The group name to extend.
<code>\$groupexpiry</code>	The expiry date. example values "30" Sets the expiry date 30 days from now "+30" Extends any existing expiry by 30 days "161209" sets the expiry date to 16th December 2009 assuming DDMMYY format is set
<code>\$expirytype</code>	This setting can be used to determine whether to extend the expiry. Here are some examples "0" Setting 0 will always adjust the expiry date "+30" Using a + in front of the number will only adjust the expiry date if the existing expiry was 30 (in this case) or more days ago. "-30" Using a - in front of the number will only adjust the expiry if there are at least 30 (in this case) days until the current expiry.
<code>\$clientemail</code>	The email template sent to the client when updated. Set to "" if not required.
<code>\$adminemail</code>	The email template sent to the admin when updated. Set to "" if not required.

After the call the function returns -1 if there was a database error, 0 if the user was not updated (usually because the user does not exist or not a member of the group) or 1 if updated successfully.

### **sl\_extendgroup**

```
$result=sl_extendgroup($group,$groupexpiry,$expirytype,  
    $clientemail,$adminemail);
```

The `sl_extendgroup` function extends the expiry date of a group for the currently logged in user. This is the same as the `slapi_extendgroup` function except that it works from a members page on the currently logged in user. The Sitelok variables and current session are updated as well.

### **sl\_ismemberof()**

```
$result=sl_ismemberof($group);
```

The `sl_ismemberof` function returns true if the currently logged in user is a member of the group (even if expired). Returns false if not a member. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. This function always returns true if the user is in ADMIN.

### **sl\_isactivememberof()**

```
$result=sl_isactivememberof($group);
```

The `sl_isactivememberof` function returns true if the currently logged in user is a member of the group (and not expired). Returns false if not a member or is expired. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. This function always returns true if the user is in ADMIN and not expired.

### **sl\_memberofexpires()**

```
$result=sl_memberofexpires($group);
```

The `sl_memberofexpires` function returns the PHP timestamp for the groups expiry date for the currently logged in user. If the user is not a member of the group then -1 is returned. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file.

## **sl\_ismemberofany()**

```
$result=sl_ismemberofany("BRONZE,SILVER,GOLD");
```

The `sl_ismemberofany` function returns true if the currently logged in user is a member of any of the groups listed (even if expired). Returns false if not a member of any. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. The parameter can also be an array of group names if preferred. This function always returns true if the user is in ADMIN.

## **sl\_ismemberofall()**

```
$result=sl_ismemberofall("SILVER,GOLD");
```

The `sl_ismemberofall` function returns true if the currently logged in user is a member of all of the groups listed (even if expired). Returns false if not a member of all of them. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. The parameter can also be an array of group names if preferred. This function always returns true if the user is in ADMIN.

## **sl\_isactivememberofany()**

```
$result=sl_isactivememberofany("BRONZE,SILVER,GOLD");
```

The `sl_isactivememberofany` function returns true if the currently logged in user is a member of any of the groups listed (and not expired). Returns false if not a member of any. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. The parameter can also be an array of group names if preferred. This function always returns true if the user is in ADMIN and not expired.

## **sl\_isactivememberofall()**

```
$result=sl_isactivememberofall("SILVER,GOLD");
```

The `sl_ismemberofall` function returns true if the currently logged in user is a member of all of the groups listed (and not expired). Returns false if not a member of all of them. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file. The parameter can also be an array of group names if preferred. This function always returns true if the user is in ADMIN and not expired.

## **sl\_setcustom()**

```
$result=sl_setcustom("1",$datatostore);
```

The `sl_setcustom` function updates the data stored in the specified custom field for the currently logged in user. The first parameter specifies which custom field to update and can be from "1" to "50". The function returns true if successful or false if a problem occurred. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file.

## **sl\_redirecttourl()**

```
sl_redirecttourl($page);
```

The `sl_redirecttourl` function causes the browser to redirect to the page specified. This uses the http header to do this. For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file.

## **sl\_getstartpage()**

```
$page=sl_getstartpage();
```

The `sl_getstartpage` function returns the login start page for this user (if one is defined). For historical reasons this function is not defined in `sitelokapi.php` so can be used without including that file.

## **slapi\_loginuser**

```
$res=slapi_loginuser($username);
```

The `slapi_loginuser` function can programmatically login a user. This should be done from a page displayed in the users browser so that the session can be started.

`$username`      Username to login

After the call the function returns 1 is successful, 0 if not and -1 if a database error occurred.

## **sl\_logout()**

```
sl_logout();
```

The `sl_logout` function can programmatically logout the current user ending the session. This should be done from a page displayed in the users browser so that the session can be ended.

The function has no return value.

## **sl\_siteloklink()**

```
$url=sl_siteloklink("text.zip",1);
```

This is the same as the `siteloklink()` function that embeds secure download links in member pages except that the URL is returned as a PHP function parameter instead of being inserted in the page. This function should be called from a members page.

## **slapi\_totalusers()**

```
$num=slapi_totalusers();
```

This function returns the total number of users in the Sitelok database.

## **slapi\_usergrouptotalusers()**

```
$num=slapi_usergrouptotalusers("GOLD");
```

This function returns the total number of users belonging to the specified usergroup.

## Chapter 17 Sitelok event handler

This chapter is for PHP developers who would like to enhance Sitelok with new features or integrate it with other systems. Most users won't need anything here but may like to read through to have an idea of what can be done.

For more advanced applications Sitelok provides special event handlers that get called when certain things happen. For example we have an event handler function that gets called whenever a new user is added to Sitelok. You could perhaps add some code to the event handler to add this same user to another system as well.

The event handler functions are all contained in the file called `eventhandler.php` in the `slpw` folder. You can add PHP code to these functions to do what you need. If you wish you can use Sitelok API function calls as well but you should add this line to the top of `eventhandler.php` to do this.

```
require_once"sitelokapi.php";
```

The Sitelok plugin system uses similar event handlers which are called separately.

### **sl\_onCustom1Validate**

This event handler function gets called whenever a `custom1` field is input and needs to be validated. The event only gets called if you have enable validation for `custom1` in the Sitelok configuration page. The default code simply checks that a value is input (i.e. for a required field) but you could use any validation you wish.

The following parameters are passed into the event handler function.

<code>\$data</code>	This is the data input into the field by the user
<code>\$title</code>	This is the title defined for custom 1 in Sitelok configuration
<code>\$from</code>	This tells you where the data is being input and allows you to handle validation differently if necessary.
<code>0</code>	Registration form
<code>1</code>	User profile form
<code>2</code>	Admin panel

The return value from the function should be a string value. If the \$data value sent in is validated then set the return string to "". If not then set the return string to the error message to display to the user.

There are fifty of these event handler functions, one for each custom field.

## **sl\_onUsernameValidate**

This event handler function gets called whenever a username field is input and needs to be validated. Return "" to allow or a string message to block.

The following parameters are passed into the event handler function.

<code>\$data</code>	This is the data input into the field by the user
<code>\$from</code>	This tells you where the data is being input and allows you to handle validation differently if necessary.
<code>0</code>	Registration form
<code>1</code>	User profile form
<code>2</code>	Admin panel

## **sl\_onPasswordValidate**

This event handler function gets called whenever a password field is input and needs to be validated. Return "" to allow or a string message to block.

The following parameters are passed into the event handler function.

<code>\$data</code>	This is the data input into the field by the user
<code>\$from</code>	This tells you where the data is being input and allows you to handle validation differently if necessary.
<code>0</code>	Registration form
<code>1</code>	User profile form
<code>2</code>	Admin panel

## sl\_onEmailValidate

This event handler function gets called whenever an email field is input and needs to be validated. Return "" to allow or a string message to block.

The following parameters are passed into the event handler function.

<code>\$data</code>	This is the data input into the field by the user
<code>\$from</code>	This tells you where the data is being input and allows you to handle validation differently if necessary.
0	Registration form
1	User profile form
2	Admin panel

## sl\_onNameValidate

This event handler function gets called whenever a name field is input and needs to be validated. Return "" to allow or a string message to block.

The following parameters are passed into the event handler function.

<code>\$data</code>	This is the data input into the field by the user
<code>\$from</code>	This tells you where the data is being input and allows you to handle validation differently if necessary.
0	Registration form
1	User profile form
2	Admin panel

## sl\_onAddUser

This event handler gets called when a new user is added to Sitelok. This could be through the admin panel, registration, Paypal module or any other means. The event is called after the user is actually added to Sitelok.

The `$data` parameter is sent into event handler function and is a string array containing all of the data for the new user as follows.

<code>\$data['username']</code>	The username
<code>\$data['password']</code>	The users password
<code>\$data['enabled']</code>	Contains "Yes" or "No" depending on whether the user is enabled or not
<code>\$data['name']</code>	The users name
<code>\$data['email']</code>	The users email address
<code>\$data['usergroups']</code>	The usergroup data for the user in the format it is stored in the database
<code>\$data['custom1']</code> to <code>\$data['custom50']</code>	The custom data
<code>\$data['from']</code>	Set to 0 is adding user from the API (including registration form) or 1 if adding user from the control panel.

There is no return parameter for this function.

## sl\_onModifyUser

This event handler gets called when a users data is modified in Sitelok. This could be through the admin panel, modify profile form, Paypal module or any other means. The event is called after the user is actually modified in Sitelok.

The `$data` parameter is sent into event handler function and is a string array containing all of the modified data for the user. This is the same as is sent to `sl_onAddUser` but also includes the following in case the username has been changed

<code>\$data['oldusername']</code>	This is the previous username
------------------------------------	-------------------------------

There is no return parameter for this function.

## **sl\_onDeleteUser**

This event handler gets called when a user is deleted from Sitelok. This could be through the admin panel or any other means. The event is usually called after the user is actually deleted but in certain circumstances this may not have been completed.

This event can get called many times when deleting selected users from the admin panel so it is important to keep your event handler code as efficient as possible.

The **\$data** parameter is sent into event handler function and is a string array containing all of the the users data. This is the same as is sent to `sl_onAddUser` which you can see for further details.

There is no return parameter for this function.

## **sl\_onLogin**

This event handler gets called when a user logs in to Sitelok.

The **\$data** parameter is sent into event handler function and is a string array containing all of the the users data. This is the same as is sent to `sl_onAddUser` which you can see for further details.

There is no return parameter for this function.

## **sl\_onLogout**

This event handler gets called when a user logs out of Sitelok.

The **\$data** parameter is sent into event handler function and is a string array containing all of the the users data. This is the same as is sent to `sl_onAddUser` which you can see for further details.

There is no return parameter for this function.

## sl\_onDownload

This event handler gets called when a user clicks a `siteloklink()` download link in a members page. It gets called before the download begins.

The following parameters are passed into the event handler function.

<code>\$user</code>	This is the username of the member performing the download
<code>\$fname</code>	This is the path and filename of the file being downloaded.
<code>\$loc</code>	This is the location name if multiple file locations are being used
<code>\$param1</code>	This is the first optional parameter that is sent in the <code>siteloklink()</code> call (see example below)
<code>\$param2</code>	This is the second optional parameter that is sent in the <code>siteloklink()</code> call (see example below)

The `$param1` and `$param2` contain the values passed in two optional parameters in the `siteloklink()` function. For example normally a `siteloklink()` looks like this in a members page.

```
<a href="<?php siteloklink('testfile.zip',1); ?>">Download here</a>
```

This would force download of `testfile.zip` when clicked.

You can add an expiry time to the link in minutes if you wish. The links only worked for the logged in user anyway so normally the default of 0 can be used.

```
<a href="<?php siteloklink('testfile.zip',1,60); ?>">Download here</a>
```

You may find it useful for some applications to pass to the event handler other data with the link which you can use `$param1` and `$param2` for. e.g.

```
<a href="<?php siteloklink('testfile.zip',1,0,"data1","data2"); ?>">Download here</a>
```

One example application for this would be if you are charging credits for downloads. You could pass in the link the credits to charge for this particular file and then have the event handler decrement these from the users existing credits stored in one of the custom fields.

There is no return parameter for this function. If during your event handler code you wish to block the download you can simply use the PHP `exit;` statement. Alternatively you could redirect the use to a certain page on the site (perhaps to explain they have no credits left in our example) and then exit.

## **sl\_onDownloadEmail**

This event handler gets called when a user clicks a `!!!link!!!` download link in an email sent out by Sitelok. It gets called before the download begins.

The following parameters are passed into the event handler function.

<code>\$user</code>	This is the username of the member performing the download
<code>\$fname</code>	This is the path and filename of the file being downloaded.
<code>\$loc</code>	This is the location name if multiple file locations are being used

There is no return parameter for this function. If during your event handler code you wish to block the download you can simply use the PHP `exit;` statement. Alternatively you could redirect the use to a certain page on the site (perhaps to explain they have no credits left in our example) and then exit.

## **sl\_onCheckLogin**

This event handler gets called after a user has entered their valid login details and gives your code a chance to block the login. Return "" to allow or a string message to display and block access.

The following parameters are passed into the event handler function.

<code>\$data['username']</code>	The username
<code>\$data['password']</code>	The users password
<code>\$data['enabled']</code>	Contains "Yes" or "No" depending on whether the user is enabled or not
<code>\$data['name']</code>	The users name
<code>\$data['email']</code>	The users email address
<code>\$data['usergroups']</code>	The usergroup data for the user in the format it is stored in the database
<code>\$data['custom1']</code>	The custom data
<code>\$data['custom50']</code>	

## sl\_onCheckRegister

This event handler gets called after a user has submitted their details and gives your code a chance to block the registration. Return "" to allow or a string message to display and block registration.

The following parameters are passed into the event handler function.

<code>\$data['username']</code>	The username
<code>\$data['password']</code>	The users password
<code>\$data['enabled']</code>	Contains "Yes" or "No" depending on whether the user is enabled or not
<code>\$data['name']</code>	The users name
<code>\$data['email']</code>	The users email address
<code>\$data['usergroups']</code>	The usergroup data for the user in the format it is stored in the database
<code>\$data['custom1']</code> to <code>\$data['custom50']</code>	The custom data

## sl\_onCheckModifyProfile

This event handler gets called after a user has submitted their details and gives your code a chance to block the update. Return "" to allow or a string message to display and block registration.

The following parameters are passed into the event handler function.

<code>\$data['username']</code>	The username
<code>\$data['password']</code>	The users password
<code>\$data['enabled']</code>	Contains "Yes" or "No" depending on whether the user is enabled or not
<code>\$data['name']</code>	The users name
<code>\$data['email']</code>	The users email address
<code>\$data['usergroups']</code>	The usergroup data for the user in the format it is stored in the database
<code>\$data['custom1']</code> to <code>\$data['custom50']</code>	The custom data

## **sl\_onPageAccess**

This function is called for every page access that Sitelok has control of (secure or PUBLIC).

The following parameters are passed into the event handler function.

<code>\$data['page']</code>	The page being accessed
<code>\$data['username']</code>	The username
<code>\$data['password']</code>	The users password
<code>\$data['enabled']</code>	Contains "Yes" or "No" depending on whether the user is enabled or not
<code>\$data['name']</code>	The users name
<code>\$data['email']</code>	The users email address
<code>\$data['usergroups']</code>	The usergroup data for the user in the format it is stored in the database
<code>\$data['custom1']</code> to <code>\$data['custom50']</code>	The custom data

There is no return value from this function.

## **sl\_onAddGroup**

This function is called when a new usergroup is added in the admin panel. The following parameters are passed in.

<code>\$data['name']</code>	The group name
<code>\$data['description']</code>	The group description
<code>\$data['loginaction']</code>	The login action
<code>\$data['loginvalue']</code>	The login value

There is no return parameter.

## sl\_onModifyGroup

This function is called when a usergroup is modified in the admin panel. The following parameters are passed in.

<code>\$data['oldname']</code>	The previous group name
<code>\$data['name']</code>	The new group name
<code>\$data['description']</code>	The group description
<code>\$data['loginaction']</code>	The login action
<code>\$data['loginvalue']</code>	The login value

There is no return parameter.

## sl\_onDeleteGroup

This function is called when a usergroup is deleted in the admin panel. The following parameters are passed in.

<code>\$data['name']</code>	The new group name
<code>\$data['description']</code>	The group description
<code>\$data['loginaction']</code>	The login action
<code>\$data['loginvalue']</code>	The login value

There is no return parameter.

## sl\_onPaypalOrder

This event handler gets called when a user has purchased a product via the Paypal Payment Module. Other events may be called as well such as `sl_onAddUser` and `sl_onModifyUser`.

The following parameters are passed into the event handler function.

<code>\$user</code>	This is the username of the buyer
<code>\$ordernumber</code>	This is the Paypal order number
<code>\$itemnumbers</code>	This is a string array of product id's ordered
<code>\$paypaldata</code>	This is an array containing all of the IPN data from Paypal
<code>\$exists</code>	This contains 1 if the user already existed or 0 if a new user

There is no return parameter for this function.

## **sl\_onUpload**

This function is called when a file is uploaded from a registration or modify profile form.

<code>\$customnum</code>	Contains the custom field number being uploaded to (1 to 50)
<code>\$fname</code>	Fullpath and filename being uploaded

There is no return parameter.

## **sl\_onPrepareEmail**

This function is called when an email is being prepared for sending. The parameters sent in are the same as for the `sl_onAddUser` function with the addition of the following specific data.

<code>\$data['subject']</code>	The email subject
<code>\$data['body']</code>	The email body
<code>\$data['htmlformat']</code>	Contains Y if email format is html
<code>\$data['toemail']</code>	The email address of the recipient

You can adjust the body and subject as required (perhaps to add custom variables etc).

There is no return parameter.

## **sl\_onSendEmailAllowed**

This function is called just before an email is sent and allows an event handler to block the sending. The same parameters are sent in as with the `sl_onPrepareEmail` function. Return true to allow the email to be sent or false to block it.

## **sl\_onSendEmailOut**

This function is called when an email is about to be sent allowing the event handler to use an alternative method to send the email. If you handle the sending then return true otherwise return false so that Sitelok can handle the email as usual.

## **sl\_onLoginFailure**

This function is called when a user fails to login because they entered incorrect credentials. The following parameters are passed in.

<code>\$data['failedtype']</code>	Set to 0 if username not recognised, 1 if the password was incorrect or 2 for extra login data incorrect.
<code>\$data['username']</code>	The username entered
<code>\$data['passwordentered']</code>	The password entered

The other user details are also provided if the username was recognized.

## Chapter 18 Support

Hopefully if you have followed this manual carefully everything will be working fine. However sometimes things don't go quite so smoothly so if you have any questions or problems then please check the FAQ on the support page or email us.

Support and updates for Sitelok are free of charge for one year after purchase. Shortly after purchase you should receive a username and password that provide you with access to the latest downloads, FAQ and other resources.

Support area: <http://www.vibralogix.com/support/>

Email: [support@vibralogix.com](mailto:support@vibralogix.com)