Lighthouse Pro

The following document explains how to install and use Lighthouse Pro. It was created on April 3rd for version 1.0. Any questions, bug reports, or suggestions, should be sent to Raymond Camden (ray@camdenfamily.com). Support forums may be found here:

http://www.coldfusionjedi.com/forums/forums.cfm?conferenceid=8AEA501F-F16B-17F0-EFB343B3B7473870

License

Installation

Extract the zip file into your web server's document root. You do not need to install it directly under web root, but these instructions will assume you have done so.

Lighthouse Pro requires Model-Glue 3. If you do not place Model-Glue under the LighthousePro directory, please be sure to create a ModelGlue mapping pointing to your installation. You can download Model-Glue 3 here: http://www.model-glue.com/

Lighthouse Pro requires the latest version of ColdSpring. As with Model-Glue, if you do not install ColdSpring under the LighthousePro directory, be sure to create a mapping for it. You can download ColdSpring here: http://www.coldspringframework.org/

Please note that if you are installing on a shared host, they may already have Model-Glue and ColdSpring installed!

If you are not using ColdFusion 8 or higher, please follow the following two steps:

1) Lighthouse Pro will require a mapping named lhp that points to the folder where you copied the files.
2) Create a new custom tag path that points to the customtags folder under your installation root.

If you are using Open BlueDragon, please follow the following two steps:

1) If you are running OpenBD on a UNIX-based system, absolute paths must begin with a "$" symbol. Therefore, in order for the "lhp" application-specific mapping to work properly, line 8 of Application.cfc must be changed to include this preceding character:

   ```
   <cfset this.mappings["/lhp"] = "$" &
   getDirectoryFromPath(getCurrentTemplatePath())>
   ```

2) Create a new custom tag path that points to the customtags folder under your installation root.

LighthousePro makes use of ColdSpring for configuration. Find the config/ColdSpring.xml.cfm file and open it with a text editor. Within the XML file you will see a block named applicationSettings. These values determine the settings for Lighthouse Pro. The following table details each setting and its meaning:

adminemail	This is the email address used for error reports and for issue notifications.
dsn	This is the datasource for Lighthouse Pro.
dbtype	This is the database type for Lighthouse Pro. It is currently unused but reserved for future use.
username	Username for datasource. Not required.
password	Password for datasource. Not required.
mailserver	Lets you specify a mail server that will override the setting in the ColdFusion administrator.
mailport	Lets you specify a port for the mail server. If left blank, it will default to 25.
mailusername	If your mail server needs a username, specify it here. You must specify both a username and a password, or leave both blank.
mailpassword	If your mail server needs a password, specify it here. You must specify both a username and a password, or leave both blank.
secretkey	This key is used to encrypt links for the RSS feed. You should modify this value to some random string. It should be at least eight characters long.

rssfeedsenabled	If true, allows users to view the RSS feeds. If false, disables the use of RSS feeds for all users.
version	Used by Lighthouse Pro to mark the version. Can be ignored.
plaintextpassword	This value determines if Lighthouse Pro will store user passwords as plain text in the database or if they will be stored as hashed values (using SHA1). Hashed passwords are more secure but prevent you from recovering what the actual password is should a user forget it. This value must be a Boolean (yes/no, 1/0, true/false) If you want to use hashed passwords for either a new or an existing installation, you must do the following: 1) set this value to "false" in defaults.cfm 2) login with an admin account 3) click the "hash passwords" link located in the admin menu once you've logged in. **Please Note**: Once you use hashed passwords, the original password values are not stored. This means you can not switch back to using plaintext passwords in the future without editing the password field in the user table manually to assign new password values for all users. The install folder contains a script to hash existing passwords. This script has a <cfabort> tag to prevent accidental usage. Remove the <cfabort>, run it, and then delete the file (you should delete the entire install folder from your server after you are done installing).

dateformat and datepickerdateformat	These two values configure how dates are formatted. Dateformat is used everywhere *except* in the JavaScript date picker. The value should be a mask that is supported by ColdFusion. The datepickerdateformat is a bit different. You can find their masks here: http://docs.jquery.com/UI/Datepicker/formatDate These two settings will allow you to use a consistent format for dates across the application.

In case you are curious – I used a CFM file instead of an XML file so I could easily protect it from being viewed by the web. Normally I'd place this XML file outside of the web root, but as an application, the entirety of Lighthouse Pro is under web root. By using a CFM file I have a simple way to protect the document.

Next you will need to determine which database you want to use. Lighthouse Pro supports SQL Server, and MySQL 4.1 or higher. Log on to your ColdFusion Administrator and create a new datasource. The name should match the value in defaults.cfm.

To initialize the database, you can either use the sqlserver.sql file or the mysql.sql file. The sqlserver.sql file is for SQL Server. The mysql.sql file is for MySQL.

Users of MySQL under ColdFusion 7 should **not** use ColdFusion's built in MySQL datasource type. Instead – follow the directions at Adobe for configuring a custom driver for MySQL:

http://www.adobe.com/go/6ef0253

Running Lighthouse Pro

At this point, everything should be setup correctly. Assuming you installed Lighthouse Pro under your web root, you can now open your browser to:

<web server>/lighthousepro

Replace <web server> with the name of your web site. If you installed Lighthouse Pro to your local server, the URL would be:

http://localhost/lighthousepro

You will be prompted to login. Out of the box, the administrator username is "admin" and the password is "password." You will want to change this immediately of course. Now let's take a look at the sections of Lighthouse Pro.

Preferences

This is where a user can modify their settings. A user can change their name, their email address, and enter a new password. Of special interest is the "Email Projects" setting. A user can elect to receive emails whenever an issue is created or updated. They select which projects they want emails for in the Preferences screen. A normal user cannot subscribe to a project he or she is not a member of. An administrator, however, can subscribe to any project.

Stats

The stats page is where you can generate reports. A user can only generate reports on the projects that they are a member of, while admin users have access to all the projects. You can also choose to generate a report on all the issues that are assigned to you. Reports can be generated with charts or in Excel format.

RSS Feeds

Each project a user has access to has a corresponding RSS feed. The RSS Feeds section lets you cut and paste URLs into your RSS reader. This lets you monitor Lighthouse Pro issues from outside the application. Please note that the URL contains your login info in an encrypted format. This allows the remote reader to gain access without forcing you to login every time. Even though the RSS feed URLs contain the user’s login info (username and password) in encrypted format, there may still be concerns about protecting the login information. You can turn off the RSS feeds entirely by setting the default variable rssfeedsenabled to false. In addition, if using hashed passwords, only the hashed password value is placed in the URL. The plain text password is never used when plaintextpassword is set to false.

Users

The users panel allows you to add, edit, and delete users. This is an admin only option. Obviously you should not delete all the admin users.

Projects

The projects panel allows you to add, edit, and delete projects. This is an admin only option. Each project has a name and a few options. The first option is the project loci. Project Loci simply identify parts of your application. So for example, a typical application has a database, a front-end, and perhaps documentation. You assign which loci belong to a project by selecting from the list of available loci.

Next you assign users to a project. Only users who are assigned to a project can add, edit, and delete issues.

Projects can have defaults. These defaults are applied to new issues and issues created via email (see below).

Projects can be setup to check an email account for issues. This lets you share an email address for folks to submit bugs. In order to use this feature, you must set four settings:

1) Mail Server - This is the mail server to connect to.
2) Mail Username - This is the username to authenticate with.
3) Mail Password - This is the password to use.
4) Mail Email Address - This is an important one. If the mail in the account does not match this email address, the mail will be ignored. This allows you to use one email account and multiple aliases to handle multiple projects. So for example, the account "bugs" could have two aliases, bugs-A and bugs-B. You can then match one project to bugs-A and another to bugs-B. While only one mail account is in use, Lighthouse Pro will check the correct email address when adding bugs.

Project Areas

The project areas panel allows you to add, edit, and delete project areas. This is an admin only option. Project Areas simply describe the areas of your project where issues may occur. Each project can assign different areas to themselves.

Issue Types

The issue types panel allows you to add, edit, and delete issue types. This is an admin only option. Issue types describe issues and help categorize them.

Statuses

This panel lets the admin define the status settings for issues.

Severities

This panel lets the admin define the severity settings for issues.

Milestones

This panel lets you create milestones for projects. Milestones are a way to group issues. So for example, you may have a Beta milestone that represents all issues that must be fixed for the beta. Simply create the milestone (with the optional due date) and then assign bugs to that milestone.

Working with Projects

Most of your time in Lighthouse Pro will involve working with projects. To work with a project, simply select it from the Projects menu. If a project has no issues, you will simply be provided with a form to add an issue. If a project does have issues, you will be presented with a grid allowing you to view, sort, and filter the issues. By selecting an issue you can see the details, and update the issue. Each issue has the following properties:

name	The name of the issue. Should be a short description of the issue. Ex: "Add User is broken."
type	Issues are classified into two simple types, bugs and enhancements.
milestone	Some issues may be assigned milestones. Milestones are created for a project.
locus	The locus of the issue, or where it is located.
attachment	A file attachment. This could be a screen shot of the error.
duedate	The due date for the issue, if any.
related url	This is a URL that points to a resource related to the issue. If you are pointing out a bug in an application, this could be the URL that will cause the error to occur.
severity	This value indicates how severe the issue is. Severity values are low, normal, and high.
status	This value indicates the current status of the issue. Possible values are: open, fixed, and closed. A developer would mark an issue fixed when they have addressed an issue. The person who created the issue would mark it closed to signify that they agree that the issue is fixed.

owner	Every issue is assigned to one developer.
created/updated	Signifies when an issue was created and when it was last updated.
description	This is a detailed description of the issue. In general, it is always best to be as descriptive as possible when creating an issue.
history	Whenever an issue is created or update, a note is added to the history. A developer can also add notes themselves. This could be used to request more information, or to clarify what an issue is really talking about. Another example would be a case where a developer marks the issue as fixed, but the create of the issue sees that the issue still exists in the application.

Announcements

Announcements allow you to tie messages to a project. This is useful for mentioning important dates or reminders for developers using Lighthouse Pro. The idea for this came from Dan Sorensen.

Filters

Filters allow you to save a current view (ie, a project with chosen issue types, keyword filters, etc). Saving a filter will save every aspect of the current view (except sort) so that you can quickly load it later.