

ELOG Electronic Logbook

None

None

None

Table of contents

1. The ELOG Home Page	5
1.1 What is ELOG ?	5
1.2 Screen shots	6
1.3 Use cases	7
1.4 License	7
1.5 Credits	7
1.6 Talks and presentations	7
2. ELOG User's Guide	8
2.1 A Quick Intro	8
2.2 What Words Mean Here	8
2.3 Accessing an ELOG server and its logbook(s)	8
2.4 Viewing information in ELOG	9
2.5 Browsing around and finding things	10
2.6 Adding stuff to a logbook	10
2.7 Misc. tips & tricks, things to be aware of...	11
2.8 elog command line client	12
2.9 RSS Feed	12
3. ELOG Administrator's Guide	15
3.1 Installing and running on Linux	15
3.2 Notes for various platforms	16
3.3 Running elogd under Apache	17
3.4 Using apache authentication	17
3.5 Installing ImageMagick	17
3.6 Installing and running in Windows	18
3.7 Server Configuration	18
3.8 Secure Connections HOWTO	19
3.9 How It All Works	19
4. ELOG - Syntax of elogd.cfg	21
4.1 Global options	21
4.2 Groups of logbooks	24
4.3 Top groups	25
4.4 Individual logbook options	26
4.5 General options	26
4.6 Attributes	32
4.7 Conditional attributes	41

4.8 Multiple conditions	42
4.9 Conditions in the list display	42
4.10 Access control	42
4.11 Password file	42
4.12 Kerberos authentication	44
4.13 LDAP authentication	45
4.14 PAM authentication	46
4.15 EMail notification	46
4.16 Flags	47
4.17 Themes	52
4.18 Mirroring	53
5. ELOG FAQ	57
5.1 How does one configure elog to display the last message by default	57
5.2 Are there any plans to implement a MySQL back end?	57
5.3 Can I run the ELOG daemon from inside Apache or any other Web server?	57
5.4 I can access my logbook without any password, isn't that a security problem?	57
5.5 I want a bookmark pointing to the last page where an attribute has a certain value	57
5.6 I want a logbook with public read access (no password), but restricted write access	58
5.7 I have many logbooks with password files, so if I add a user or want to change a password I have to do this for all logbooks which is painful.	58
5.8 How can I configure ELOG such that it displays something else than the message list by default?	58
5.9 I want to have additional commands specific to my lookbook	58
5.10 How does one configure elog to disable editing of existing messages? I want a logbook where one can enter messages but not change them afterwards.	58
5.11 How can I track various revisions of a message using the "edit" command?	59
5.12 How can I enter a date which is different from current one?	59
5.13 I cannot pass the login page, it's always redisplayed even if I put in the right password?	59
5.14 How can I change an attribute for an entire thread? We have an attribute "open problem/fixed" which should be changed for the whole thread if that problem has been fixed.	59
5.15 Can I use RSS feeds with password protected logbooks?	60
5.16 How can I make a whole thread open or closed?	60
5.17 Does elog have a spell checker?	60
5.18 Why are entries with large attachments submitted so slowly?	60
5.19 The elgod daemon crashes from time to time, what can I do?	61
5.20 How can I create an ELOG entry automatically from a script?	61
5.21 I want to notify different people for different things, how do I set up this?	61
6. Forum	62
7. ELOG Wishlist	63
8. ELOG Contributions	64

9. Demo	65
10. ELOG download information	66
10.1 Binary packages	66
10.2 Windows Binaries	66

1. The ELOG Home Page

Home of the *Electronic Logbook* package by [Stefan Ritt](#)

Current version: 3.1.4

[PDF version](#)

1.1 What is ELOG ?

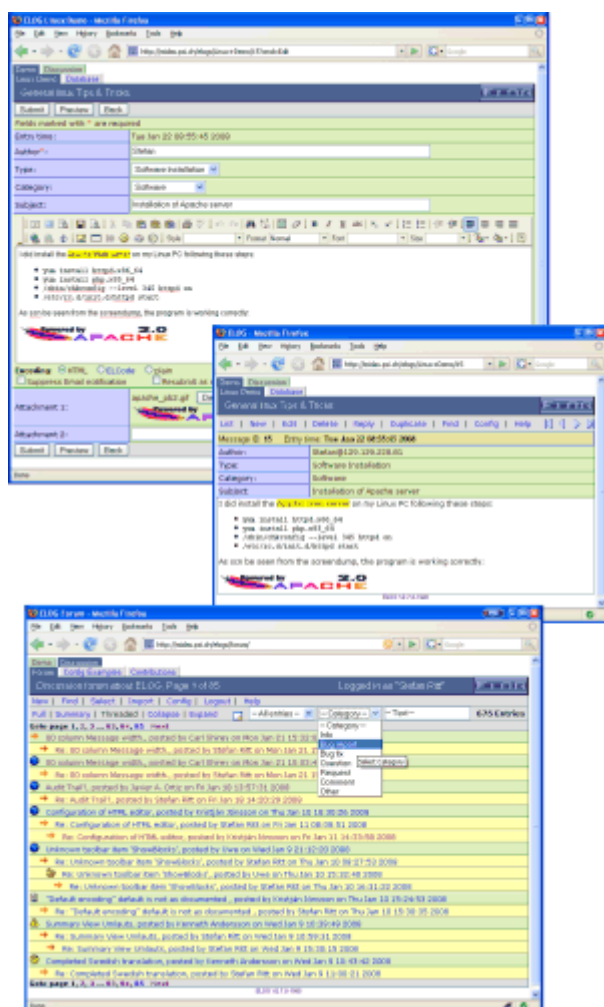
ELOG is part of a family of applications known as *weblogs* . Their general purpose is :

1. to make it easy for people to put information online in a chronological fashion, in the form of short, time-stamped text messages ("entries") with optional HTML markup for presentation, and optional file attachments (images, archives, etc.)
2. to make it easy for other people to access this information through a Web interface, browse entries, search, download files, and optionally add, update, delete or comment on entries.

ELOG is a remarkable implementation of a *weblog* in at least two respects :

- its simplicity of use : you don't need to be a seasoned server operator and/or an experimented database administrator to run **ELOG** ; one executable file (under Unix or Windows), a simple configuration text file, and it works. No Web server or relational database required. It is also easy to translate the interface to the appropriate language for your users.
- its versatility : through its single configuration file, **ELOG** can be made to display an infinity of variants of the *weblog* concept. There are options for what to display, how to display it, what commands are available and to whom, access control, etc. Moreover, a single server can host several *weblogs*, and each *weblog* can be totally different from the rest.

1.2 Screen shots



On the left upper panel is a typical logbook page displayed by Netscape Navigator. Each logbook page can contain attachments in a similar way to emails. This makes it possible to store images or text files and retrieve them easily. You could for example attach a sample configuration file which can later be copied to the local machine with the "Save As..." function of the Web browser.

Several logbooks can be served through a single **ELOG** server. Each logbook can use different attributes for its entries. The logbook can then be searched using these attributes. The right pane on the left image shows a search for all entries with attribute "Type" equal to "Configuration", and the lower pane shows the search result. It is also possible to use full-text search in attributes and the entry body.

While logbook entries are usually displayed one entry per page, they can also be listed consecutively which makes it easy to produce a paper printout of a logbook.

Logbook pages can be edited or deleted. This feature can be turned off in the configuration file so that a logbook entries cannot be changed after being submitted.

An additional feature is the automatic generation of a notification email messages based on a certain type or category of a logbook entry.

Also try out the [online demo](#)

1.3 Use cases

The features of **ELOG** make it useful for several applications:

- **Personal Logbooks.** Personal notes can be written into **ELOG** and can then be retrieved from anywhere with a Web browser. This makes it handy for PC supporters who have to go around in companies or laboratories and don't want to carry their paper logbook with them. The same holds true for people traveling around a lot. The logbook database consists of plain ASCII files which can be copied easily between different computers to have local access, for example on a notebook with no network connection.
- **Shared Logbooks.** Logbooks can be shared by several people, for reading and optionally for writing. This way workgroups can share and exchange information like in a (simplified) news group. This is supported by the *Reply* command in **ELOG** which creates "*threads*" of entries. Users can be notified by email when new entries are added to the logbook. Compared to that of a news server, the installation of **ELOG** is much simpler.
- **Small Databases.** Since arbitrary attributes can be defined for a logbook, it can be used as a small database with search facilities.
- **Problem collections.** A system can consist of two logbooks, in one of which users enter bugs or problems. If someone adds a problem, an email is automatically sent to the administrator, who can then copy the entry to the second logbook and add the solution to the problem. Users can then look up all fixed problems.
- **Shift Logbooks.** If the *Allow delete* and *Allow edit* flags are off, an entry cannot be modified once it's been entered. This can be useful for shift logbooks for example in accelerator control rooms where each entry becomes a "*document*" with a time and author stamp. **ELOG** was originally developed as a shift logbook for the [PiBeta](#) and [Muegama](#) particle experiments at [PSI](#).
- **File collections.** Since files can be attached to **ELOG** entries, the system can be used to store and retrieve files. This can be used to store configuration files, which need to be accessible by several people over the web, or to store images. Since **ELOG** features an elaborate query facility, entries can be searched for by specifying several categories.

1.4 License

ELOG is released under the [GNU Public License](#) .

1.5 Credits

The author would like to give credits to following people:

- [Fred Pacquier](#) for this Web site and the French translation
- [Recai Oktas](#) and [Roger Kalt](#) for the Debian package
- [djek](#) for the Dutch translation
- [Heiko Scheit](#) for many bug fixes and fruitful discussions
- [Julio Calvo](#) for the Spanish translation
- [Emiliano 'AlberT' Gabrielli](#) for his idea of scaling attached images
- [Andreas Luedeke](#) for continuing user support and deployment of ELOG at PSI

1.6 Talks and presentations

Here are some talks and presentations given at various occasions:

- Seminar at KIT, Karlsruhe, Jan. 2015. [Introduction talk](#) by Stefan Ritt.
- Seminar at KIT, Karlsruhe, Jan. 2015. [Application of ELOG](#) for accelerator operation at PSI by [Andreas Luedeke](#).

Content by [Stefan Ritt](#), Web pages by [Fred Pacquier](#)

2. ELOG User's Guide

How to get the most from your ELOG server

2.1 A Quick Intro

ELOG is part of a family of applications known as *weblogs*. Their general purpose is :

1. to make it easy for people to put information online in a chronological fashion, in the form of short, time-stamped text messages ("*entries*") with optional HTML markup for presentation, and optional file attachments (images, archives, etc.)
2. to make it easy for other people to access this information through a Web interface, browse entries, search, download files, and optionally add, update, delete or comment on entries.

ELOG is a remarkable implementation of a *weblog* in at least two respects :

- its simplicity of use : you don't need to be a seasoned server operator and/or an experimented database administrator to run **ELOG** ; one executable file (under Unix or Windows), a simple configuration text file, and it works. No Web server or relational database required. It is also easy to translate the interface to the appropriate language for your users.
- its versatility : through its single configuration file, **ELOG** can be made to display an infinity of variants of the *weblog* concept. There are options for what to display, how to display it, what commands are available and to whom, access control, etc. Moreover, a single server can host several *weblogs*, and each *weblog* can be totally different from the rest.

This is actually a problem when writing a User's Guide, because **ELOG** servers, and individual **weblogs** on one server, can vary wildly in appearance and functionality... This guide only attempts to cover the main concepts of importance for **ELOG** users, describing the default "*out-of-the-box*" setup and how that behaviour may have been modified by the server administrator.

2.2 What Words Mean Here

Just to be clear, some definitions of terms that will be used throughout the guide :

- **ELOG server** : the machine on which the **ELOG** server is run. Its operating system (Windows/Unix/Linux) and status (server/desktop) are not important, and of course it will probably do many other things besides.
- **ELOG administrator** : the person who has the authority to modify the **ELOG** configuration file on the server. May be an actual system administrator, a normal user of a server, or just the owner of a Windows PC.
- **logbook** : a *weblog* made available by the **ELOG** server. There may be many distinct such logbooks on one server.
- **entry** : the individual piece of information in a logbook. Can be as basic as a text message with a time-stamp, or carry much more information : attributes (see below), HTML markup, links, attached files...

2.3 Accessing an ELOG server and its logbook(s)

To access a logbook, point your Web browser at the appropriate URL. The default for a local Elog is `http://localhost:8080/Logbookname`. Logbook files are stored in directory `Logbookname` which is a sub-directory of the logbook root directory, defined by the administrator. See the administrator guide on how to create a new logbook.

If several logbooks are defined on the server, the entry page may be a list of all logbooks, with their descriptions, number of entries, and links to enter the logbook you want to use.

Alternatively, you may be taken directly to a specific logbook. By default you will see a list of entries, but the administrator may have defined a different "*default view*" for the logbook, like the list of the day's entries, or directly display the last entry, etc. (depending on what is most convenient for that logbook's purpose).

Each entry in a logbook is identified by a unique ID, which is the last part of the URL when that message is displayed. This ID might be used to create a bookmark in a browser pointing directly to a specific entry.

There are four ways through which access to a logbook may be controlled: it may be open for all to read ; it may require a common "*read*" password for all users ; it may require each user to have an individual user account (login name) and password ; finally, access may be granted or not depending on the address of the workstation you are using.

2.4 Viewing information in ELOG

There are two main viewing modes in a logbook :

- the "**entry**" view : this is when only one entry is displayed on screen (like the latest entry when you first enter a logbook, or if you click on one in a list). Here are the various parts of the display :
 - if there are several logbooks on the **ELOG** server you will see a row of "*tabs*" at the top with the names of all the logbooks. These are links that allow to switch quickly between logbooks (*this may be disabled*).
 - below is a title bar with the name of the current logbook at the left, and the **ELOG** logo at the right. If you are logged in, there will be a "Logged in as <username>" reminder in between.
 - next is the "*menu bar*" : on the left is a series of links or buttons for **ELOG** commands available to you. These are explored in the sections below (*Note: different users may see different menus*). On the right is a "*VCR-like*" set of buttons for browsing, also explained later (*this may be disabled*).
 - after these comes the actual entry information. It always starts with the entry time-stamp, and may be followed by up to twenty "*attributes*". These are like fields in a database and have been defined specifically for the current logbook. Each attribute has a checkbox besides it, explained below (*this may be disabled*).
 - the full-width box below holds the textual content (message) of the entry. This can be plain-text or HTML code. Note that for some special applications (say, a photo album or an event log) the attributes and/or the attached files may be enough information, so this field may not always be present.
 - last and optionally, one or more attached files (that were uploaded to the server when the entry was created) are offered as clickable links for download or viewing, along with the file name and size. If these are images they may be displayed directly on the page.
- At the bottom of every page is a common "*footer*" for the logbook. By default this is just a link to the **ELOG** home page in Switzerland, but may be customized locally (typically to provide a navigation bar and links for integration with other Web sites).
- the "**search result**" views : these are basically lists of entries, resulting either from a "*Search*" command or from shortcuts such as "*Last X days*" and "*Last X entries*" commands (more on this below). This mode has many options, including : - a "*summary*" view : one entry per row in a table. Some attributes may not be displayed. If the entry text is displayed (or its first few lines), it goes into the rightmost column. Attachments are not displayed.
 - a "*classical weblog*" view : entries appear beneath one another, with attributes on one line and the text (and attachments, if present) below. Images may be displayed or just linked to.
 - entries may appear most recent first, or in reverse.
 - menus on list views are different from the entry view menu. By default they only have two or three commands, but they may have been customized by the administrator to add more.

All these lists have a number to the left of each listed entry, that is a link to the corresponding entry view.

2.5 Browsing around and finding things

There are several interesting ways to peruse the information in a logbook :

- **weblogs** are often used for applications where chronology (time) is relevant, so a very common approach is to see "*what happened last*". In **ELOG** there are two commands for this. They are actually shortcuts for searches, to display the last day's (24 hrs) entries, or the last 10 entries (regardless of age). Note that the menus on the "*search result*" views of these commands are a bit special : they have the same command that created them, but with the search "*interval*" doubled. From the "*last day*" list you can get the "*last 2 days*" list, from that one the "*last 4 days*", etc., and similarly for "*last 10*", "*last 20*", etc., making it easy to quickly go back in time.
- another useful method, very specific to **ELOG**, is "*filtered browsing*" - again, shortcuts for specific searches. On the entry view, the "*VCR*" buttons normally let you see the previous, next, first or last entry in the logbook. However, if on the current entry you check one (or more) of the checkboxes in front of the attributes, only entries having the same value for the checked attribute(s) will be displayed by the browse buttons. Thus you can quickly flip through all the entries you submitted yourself, or of a certain type/category, depending on what attributes have been defined.
- for custom searches there is the query form given by the "*Find*" command. This lets you look for entries between two dates, with particular values for any attribute, or containing specific text. If you fill in several fields, only entries that meet **ALL** criteria will be selected. Possible options include sort order and summary view for results, printer-friendly formatting, displaying attachments or not, and searching through all logbooks on the **ELOG** server (if applicable).

2.6 Adding stuff to a logbook

If you have "*write access*" to a logbook (by one of the same four methods as for read access), then you may use the "*New*", "*Edit*", "*Reply*" and "*Delete*" commands.

For the quality of the information committed to the logbook, you need understand and use these as well as possible. Here are some of the important features for each command :

• **New :**

- you will not be able to save your entry if all attributes marked with a red star (*) are not filled in.
- some attributes may be pre-filled from system variables (like your user name). Pre-filled attributes may be still editable or read-only (like the entry creation date).
- attributes may be text fields (limited to 100 characters), list-boxes (max. 100 values), or check-boxes. There is also a special type of attribute where several values are listed on a line with check-boxes, and you can check as many values as needed.
- a nice touch : URLs in attributes (<http://...>, <ftp://...>, <mailto:...>) are automatically converted to links.
- in addition to the above URLs, one can enter a tag **elog:<id>** which references another logbook entry. The tag **elog:<logbook>/<id>** references a message in another logbook on the same server. The tag **elog:<id>/<n>** references attachment number **n** in a logbook entry. To reference an attachment in the current message, one uses **elog:<n>**. An anchor inside an entry can be referenced with **elog:<id>#<anchor>**.
- the Text multi-line field, if present, may be pre-filled with a template if entries need to have a common, consistent format across the logbook (especially for HTML). There may also be a comment inserted before it to explain local rules and conventions, upload rules, etc.
- check the "Submit as HTML" box if the entry contains HTML markup.
- a logbook may be configured to send a notification e-mail to various recipients each time an entry is submitted. This may be the default behaviour, and you should check "Suppress notification" if it is not wanted. Or it may be checked by default, and you need to explicitly uncheck it to send the mail. Then again, you may not have a choice... (note that notification recipients may or may not be disclosed).
- if the logbook allows attachments, there will be a number of fields with "Browse" buttons at the bottom of the form. Use these to pick one or more files on your local computer, they will be uploaded to the **ELOG** server as you submit the form. IMPORTANT : there is an upper limit on the size of individual attached files. By default it is about 1 MB but can be changed by the administrator.

• **Edit :**

- normally the Edit form will have all the values of the existing entry in its fields for modification. However, sometimes you may see fields that have been blanked if this makes sense for a particular logbook application (e.g. a "Last modified by" field).
- the "Submit as new entry" checkbox only appears on Edit forms. If it is unchecked, the modified entry keeps its original creation time-stamp. If it is checked, the modified entry becomes the latest in the logbook, as if it had just been created. Again, it is possible that this is checked by default, or disabled altogether on some logbooks.
- managing attachments through this form is easy. If all you want to change is the attributes or text, don't touch the fields at the bottom and the original attachments will be preserved. If you want to add an additional attachment, use an empty field. If you want to update an existing file, use the "Browse" button below that file's name to specify the new one. Lastly, if you want to delete an attachment without uploading a new one in its place, you must type the magic word "<delete>" in the field below its name.

• **Reply :**

- this command creates a new entry, but with the current entry's text "quoted" (with \>\) in the compose form, much like when replying to e-mail.
- the new entry has a special "In reply to" attribute with a link to the original entry ; the latter also acquires a "Reply" attribute with a link to the new entry. Unfortunately these links cannot be trusted in the present **ELOG** storage system, and the whole scheme gets somewhat confusing when there are several replies.

• **Delete :**

- nothing much to say about this one, except that there is no "Recycle bin" or whatever : once you have confirmed the deletion of an entry, it's gone for good, so be careful ! (same holds for the replacement or deletion of an attached file).

2.7 Misc. tips & tricks, things to be aware of...

- you can link directly to a specific entry by its URL, using the message ID (from another entry or an external Web page). It is also possible to link to a search result this way: use the "Search" form to compose a query that will result in exactly what you want (either a single entry or a list of entries). Copy the URL for that result page from your browser, and use that as the target for your link.
- right now you cannot search entries for attachments by their file name.
- right now attributes that consist of just a checkbox ("boolean") can only be searched by "checked" state in the "Search" form. However, if you start from an entry where that attribute is unchecked, you can use "filtered browsing" to flip through all other entries where it is also unchecked.

as mentioned above, the *"Reply"* command only provides a basic comment/chat facility - a full-blown discussion board is not **ELOG**'s purpose. If a logbook has a very specific purpose and format (picture gallery, event log, file library etc.) it might be a good idea to disable that command there and move all chat/comments/discussions to a separate, dedicated logbook to avoid *"visual pollution"*.

- it is important to understand that currently the **ELOG** server application is *"single-process"* and *"non-streaming"*. In normal terms this means that :
- only one request is processed at any one time by the server.
- uploading or downloading an attachment file is a single request, and causes the entire file to be loaded in server memory while the request is being processed.

This is not normally a problem for the sort of short, text-mode entries **ELOG** is designed to support. However, if a user starts to upload or download a large attachment file (or image) over a slow link, all other users on that **ELOG** server will have to wait for that transfer to finish before they can access any logbook on that server. This is why there is a low limit on the size of attachments, and why **ELOG** should not be used to distribute large files under intensive multi-user conditions.

- It is possible to use bookmarks to pre-populate various attributes when submitting an **ELOG** entry. This can be useful if the same person often creates similar entries from the same PC. For example, with a bookmark of the form:

`http://your.host/your_logbook/?cmd=New&pauthor=joe&ptype=Info`

...a new entry is created, with the *"author"* field pre-populated with *"joe"* and the *"Info"* value preselected for the *"type"* field. The same is possible for any attribute defined in the logbook (note the leading *"p"*). Thus you can define a set of bookmarks for various types of logbook entries.

2.8 elog command line client

In addition to submission of logbook entries through the Web interface, the standalone *"client"* program **eLog** can be used.

The parameters are:

```
elog <parameters>

-h <hostname>      Hostname where elogd is running
[-p port]          Port where elogd is running
[-d subdir]        URL Directory where elogd is running
-l logbook         Name of logbook
-s                Use SSL for communication
[-v]              For verbose output
[-w password]      Write password defined on server
[-u username password] User name and password
[-f <attachment>] Up to 50 attachments
-a <attribute>=<value> Up to 50 attributes
[-r <id>]          Reply to existing message
[-q]              Quote original text on reply
[-e <id>]          Edit existing message
[-x]              Suppress email notification
[-n 0|1|2]         Encoding: 0:ELcode,1:plain,2:HTML
-m <textfile> | <text>
```

Arguments with blanks must be enclosed in quotes. The elog message can either be submitted on the command line, piped in like

```
cat text | elog -h ... -l ... -a ...
```

or in a file with the **-m** flag. Multiple attributes and attachments can be supplied. If attributes with multiple possible values are defined in a logbook (via the *"MOptions"* keyword), they can be separated with a *"|"*, like **-a "<attribute>=<value1> | <value2>"**. The message text can be supplied directly at the command line or submitted from a file with the **-m** flag.

The **eLog** program makes it possible to submit logbook entries automatically by the system or from scripts. In some shift logbooks this feature is used to enter alarm messages automatically into the logbook.

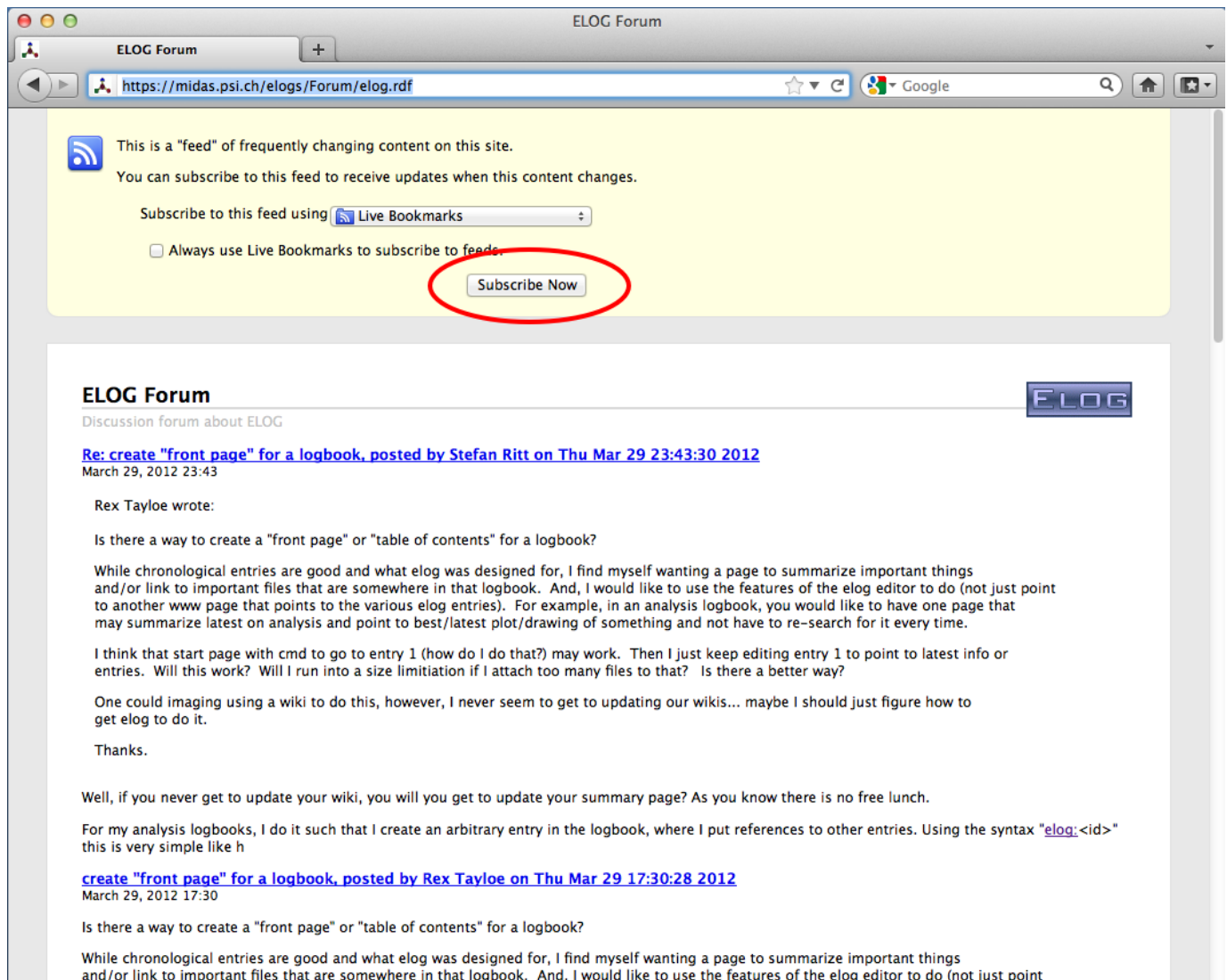
2.9 RSS Feed

RSS (RDF Site Summary or Really Simple Syntication) is a web feed format to publish frequently new or updated ELOG entries. This is a bit like the email notifications present in ELOG, but the RSS system does not go through an email reader, but through a dedicated RSS reader. This helps to separate ELOG updates from other email or spam. An RSS "channel" can be subscribed to, so one gets notified whenever a new or updated entry exists. One can either use a dedicated RSS reader or aggregator, or use the RSS functionality of a web browser, such as Firefox or Google Reader.

To obtain the RSS feed, one simply has to request the file **eLog.rdf** from a logbook. For the ELOG forum, one can enter the URL

<https://elog.psi.ch/elog/Forum/elog.rdf>

The browser then offers the possibility to subscribe to that logbook:



In case of "Live Bookmarks" in Firefox, new logbook entries automatically appear in the bookmark list:

The screenshot shows a web browser window with the title 'ELOG'. The address bar displays 'https://midas.psi.ch/elog/Forum/67223'. The page content includes a navigation menu with links like 'Demo', 'Discussion', 'Forum', 'Config Examples', and 'Contributions'. A list of forum topics is visible, including 'Re: create "front page" for a logbook, posted by Stefan Ritt on Thu Mar 2...' and 'Re: create "front page" for a logbook, posted by Rex Tayloe on Thu Mar 29 1...'. A detailed view of a post by Rex Tayloe is shown, with the subject 'Re: create "front page" for a logbook, posted by Rex Tayloe on Thu Mar 29 1...'. The post content discusses creating a 'front page' or 'table of contents' for a logbook. A context menu is open over the post, showing options like 'Open "ELOG Forum"' and 'Open All in Tabs'.

Standalone RSS reader can also notify the user of new entries with dialog boxes and sounds. For a list of available RSS aggregators, see [here](#).

3. ELOG Administrator's Guide

How to set up and run your very own ELOG server

3.1 Installing and running on Linux

3.1.1 Installation from the RPM file

Since version 2.0, ELOG contains a RPM file which eases the installation. Get the file `elog-x.x.x-i386.rpm` from the [download](http://elog.psi.ch/elog/download/RPMS/) section and execute as root `"rpm -i elog-x.x.x-i386.rpm"`. This will install the `**elogd**` daemon in `/usr/local/sbin` and the `elog` and `elconv` programs in `/usr/local/bin`. The sample configuration file `elogd.cfg` together with the sample logbook will be installed under `/usr/local/elog` and the documentation goes to `/usr/share/doc`. The `elogd` startup script will be installed at `/etc/rc.d/init.d/elogd`. To start the daemon, enter `

- `/etc/rc.d/init.d/elogd start`

It will listen under the port specified in `/usr/local/elog/elogd.cfg` which is 8080 by default. So one can connect using any browser with the URL:

- `http://localhost:8080`

To start the daemon automatically, enter:

- `chkconfig --add elogd` `chkconfig --level 345 elogd on`

which will start the daemon on run levels 3,4 and 5 after the next reboot.

Note that the RPM installation creates a user and group `elog`, under which the daemon runs.

To start the daemon on non-RedHat systems, like SuSE or Solaris, a more generic startup scrips has been provided by Steve Jones in the [Contributions](#) section.

3.1.2 Installation from the tarball

[Download](#) the latest `elog-x.x.x.tar.gz` package.`

Make sure you have the `libssl-dev` package installed. Consult your distribution for details.

Expand the compressed TAR file with `tar -xvzf elog-x.x.x.tar.gz`. This creates a subdirectory `elog-x.x.x` where x.x.x is the version number. In that directory execute `make`, which creates the executables `elogd`, `elog` and `elconv`. On some systems like OpenBSD you have to execut `gmake`. These executables can then be copied to a convenient place like `/usr/local/bin` or `~/bin`. Alternatively, a `"make install"` will copy the daemon `elogd` to `SDESTDIR` (by default `/usr/local/sbin`) and the other files to `DESTDIR` (by default `/usr/local/bin`). These directories can be changed in the Makefile. The `elogd` executable can be started manually for testing with :`

```
elogd -p 8080
```

where the `-p` flag specifies the port. Without the `-p` flag, the server uses the standard WWW port 80. Note that ports below 1024 can only be used if `elogd` is started under root, or the `"sticky bit"` is set on the executable.

When `elogd` is started under root, it attaches to the specified port and tries to fall-back to a non-root account. This is necessary to avoid security problems. It looks in the configuration file for the statements `Usr` and `Grp`. If found, `elogd` uses that user and groupe name to run under. The names must of course be present on the system (usually `/etc/passwd` and `/etc/group`). If the statements `Usr` and `Grp` are not present, `elogd` tries user and group `elog`, then the default user and group (normally `nogroup` and `nobody`). Care has to be taken that `elogd`, when running under the specific user and group account, has read and write access to the configuration file and logbook directories. Note that the RPM installation automatically creates a user and group `elog`.

If the program complains with something like `"cannot bind to port..."`, it could be that the network is not started on the Linux box. This can be checked with the `/sbin/ifconfig` program, which must show that `eth0` is up and running.

The distribution contains a sample configuration file `eLogd.cfg` and a demo logbook in the `demo` subdirectory. If the `eLogd` server is started in the `elogd-x.x.x` directory, the demo logbook can be directly accessed with a browser by specifying the URL `http://localhost:8080` (or whatever port you started the elog daemon on). If the `eLogd` server is started in some other directory, you must specify the full path of the `eLogd` file with the `"-c"` flag and change the `Data dir =` option in the configuration file to a full path like `/usr/local/elog`.

Once testing is complete, `eLogd` will typically be started with the `-D` flag to run as a *daemon* in the background, like this :

```
eLogd -p 8080 -c /usr/local/elog/eLogd.cfg -D
```

Note that it is mandatory to specify the full path for the `eLogd` file when started as a daemon.

To test the daemon, connect to your host via :

```
http://your.host:8080/
```

If port 80 is used, the port can be omitted in the URL. If several logbooks are defined on a host, they can be specified in the URL :

```
http://your.host/<logbook>
```

where `<logbook>` is the name of the logbook.

The contents of the all-important configuration file `eLogd.cfg` are described in [Config](#).

3.2 Notes for various platforms

This section contains notes for installing and running elog under various operating systems.

3.2.1 Mac OS X

Under Mac OSX, **ELOG** must be compiled from the source code. The OSX command line tools (compiler & Co) must be available, which can be done through the free Xcode package which can be obtained through the App Store. Once Xcode is installed, you can do a `xcode-select --install` to install the command line tools. After that, a simple `make` in the `elog` directory does the job of compiling ELOG. If SSL support is needed (access via `https://...`), you have to install OpenSSL and turn on SSL support in the Makefile by setting `USE_SSL = 1`. You can install OpenSSL for example through the [MacPorts](#) project. After having installed MacPorts, you do a `sudo port install openssl`.

After successful compilation, you do a `sudo make install` to install all required files under the installation directory, which is by default `/usr/local/`. A subdirectory `/usr/local/elog` is created which contains a simple example logbook. The ELOG server can now be started either manually with

```
/usr/local/sbin/eLogd
```

or through the daemon services with

```
sudo launchctl enable system/ch.psi.eLogd
sudo launchctl bootstrap system /Library/LaunchDaemons/ch.psi.eLogd.plist
```

To stop the service, use

```
sudo launchctl bootout system /Library/LaunchDaemons/ch.psi.eLogd.plist
sudo launchctl disable system/ch.psi.eLogd
```

3.2.2 Debian

A Debian package is available under <https://tracker.debian.org/pkg/elog>.

3.2.3 Solaris

[Martin Huber](#) reports that under Solaris 7 the following command line is needed to compile `elog`:

```
gcc -L/usr/lib/ -ldl -lresolv -lm -ldl -lnsl -lsocket eLogd.c -o eLogd
```

With some combinations of Solaris servers and client-side browsers there have also been problems with **ELOG's** *keep-alive* feature. In such a case you need to add the `"-k"` flag to the `eLogd` command line to turn keep-alives off.

3.2.4 FreeBSD

David Otto maintains the [ELOG port for FreeBSD](#). To install ELOG on a FreeBSD system, you can simply type

```
cd /usr/ports/www/elog
make install clean
```

3.3 Running elogd under Apache

For cases where **elogd** should run under port 80 in parallel to an Apache server, Apache can be configured to run Elog in a subdirectory of Apache. Start **elogd** normally under port 8080 (or similarly) as noted above and make sure it's working there. Then put following redirection into the Apache configuration file:

```
Redirect permanent /elog http://your.host.domain/elog/
ProxyPass /elog/ http://your.host.domain:8080/
```

Make sure that the Apache modules `mod_proxy.c` and `mod_alias.c` are activated. Justin Dieters <enderak@yahoo.com> reports that `mod_proxy_http.c` is also required. The *Redirect* statement is necessary to automatically append a "/" to a request like `http://your.host.domain/elog`. Apache then works as a proxy and forwards all requests starting with `/elog` to the `elogd` daemon.

Note: Do not put "ProxyRequests On" into your configuration file. This option is not necessary and can be misused for spamming and proxy forwarding of otherwise blocked sites.

Because **elogd** uses links to itself (for example in the email notification and the redirection after a submit), it has to know under which URL it is running. If you run it under a proxy, you have to add the line:

```
URL = http://your.proxy.host/subdir/
```

into `elogd.cfg`.

3.4 Using apache authentication

It is also possible to login via an apache-auth module. In `elogd.cfg` you should use the keyword "Webserver" for Authentication:

```
Authentication = Webserver
```

This triggers `elogd` to use the environment variable "X-Forwarded-User" as the logged in user. A simple example of a apache configuration (including the proxy) is :

```
# this required to pass on the generated env-variable X-Forwarded-User to the proxy
ProxyPassInterpolateEnv On
```

```
ProxyPass /elog/ http://your.host.domain:8080/
```

```
<Location "/elog">
    Order allow,deny
    Allow from all
    AuthType Basic
    AuthName "elog-server"
    AuthUserFile "/opt/elog/htpasswd"
    require valid-user
    RequestHeader unset Authorization
    RequestHeader add X-Forwarded-User %{REMOTE_USER}s
    # elog doesn't like the '@', so we need to cut it
    RequestHeader edit X-Forwarded-User "@(.*)$" ""
</Location>
```

3.5 Installing ImageMagick

When images are attached to ELOG entries, thumbnails can be created for quick preview. This works also for PDF and PostScript files. ELOG forwards any image operation to the ImageMagic and GhostScript packages, which must be installed for this to work. While these packages are installed on most Linux systems, windows users have to download and install these packages manually. ImageMagick can be obtained from www.imagemagick.org

[emagick.org](http://www.imagemagick.org) and GhostScript can be obtained from <http://pages.cs.wisc.edu/~ghost/>. After the installation, it has to be made sure that both packages are in the path. This can be checked to open a command prompt and typing `identify -version`. This command should return something like:

```
C:\>identify -version
Version: ImageMagick 6.3.8 01/25/08 Q16 http://www.imagemagick.org
Copyright: Copyright (C) 1999-2008 ImageMagick Studio LLC
```

When ELOG is started interactively, it checks for the ImageMagick installation and shows a note if it is found:

```
C:\Program Files\ELOG>elogd
elogd 2.7.2 built Feb 21 2008, 20:00:42 revision 2051
ImageMagick detected
Indexing logbooks ... done
Server listening on port 8080 ...
```

If ImageMagick is not installed, the thumbnail functions are simply disabled, but ELOG can otherwise run normally.

3.6 Installing and running in Windows

ELOG is distributed in binary (executable) form for Windows platforms. It will run happily in *console mode* (or "DOS box") under Windows 9x and ME. Under Windows NT and 2000 it is also possible to run it as a *service* (the Windows equivalent of a UNIX *daemon*).

[Download](#) the latest `eLogxxx.exe` file and execute it. The installer puts the **ELOG** system into a directory you specify and adds some menu shortcuts. With these shortcuts, the daemon `elogd.exe` can be started directly and the demo logbook can be accessed with the browser. Alternatively, the `elogd.exe` daemon can be registered as a service under Windows NT/2000/XP, so it gets started automatically when windows boots. This can be selected during installation or be done manually with the start menu shortcuts.

While the pre-2.5.3 methods of installing `elogd.exe` as a daemon (namely FireDaemon and `srvany.exe`) are still possible, they are not recommended any more.

Under Windows, the ports below 1024 can be used without restriction. So if no web server is running on the same PC the **ELOG** daemon can be started under the standard Web port 80. This is achieved by changing the `port=8080` option in `elogd.cfg` to `port=80` and restarting `elogd`.

3.7 Server Configuration

[The **ELOG** daemon `elogd` can be executed with the following options :]{#config}

```
elogd [-p port] [-n hostname/IP] [-C] [-m] [-M] [-D] [-c file] [-s dir] [-d dir] [-v] [-k] [-f file] [-x]

with :

`-p <port>` TCP port number to use for the http server (if other than 80)
`-n <hostname or IP address>` in the case of a "multihomed" server, host name or IP address of the interface ELOG should run on
`-C <url>` clone remote elogd configuration
`-m` synchronize logbook(s) with remote server
`-M` synchronize with removing deleted entries
`-l <logbook>` optionally specify logbook for -m and -M commands
`-D` become a daemon (Unix only)
`-c <file>` specify the configuration file (full path mandatory if -D is used)
`-s <dir>` specify resource directory (themes, icons, ...)
`-d <dir>` specify logbook root directory
`-v` verbose output for debugging
`-k` do not use TCP keep-alive
`-f <file>` specify PID file where elogd process ID is written when server is started
`-x` enables execution of shell commands
```

The appearance, functionality and behaviour of the various logbooks on an **ELOG** server are determined by the single `elogd.cfg` file in the **ELOG** installation directory.

This file may be edited directly from the file system, or from a form in the **ELOG** Web interface (when the *Config* menu item is available). In this case, changes are applied dynamically without having to restart the server. Instead of restarting the server, under Unix one can send a HUP signal like `"killall -HUP elogd"` to tell the server to re-read its configuration.

The many options of this unique but very important file are documented on the separate [elogd.cfg syntax page](#).

To better control appearance and layout of the logbooks, `elogd.cfg` may optionally specify the use of additional files containing HTML code, and/or custom "*themes*" configurations. These need to be edited directly from the file system right now.

The meaning of the directory flags `-s` and `-d` is explained in the section covering the configuration options `Resource dir` and `Logbook dir` in the [elogd.cfg description](#).^{**}

3.8 Secure Connections HOWTO

3.8.1 Using elogd itself

Starting from version 2.7.3 on, the `elogd` program supports secure connections over the Secure Socker Layer (SSL) directly. **It is recommended to run elog only through secure HTTPS connections if passwords are used. Otherwise the passwords are send over the network in clear text and exposed to sniffing attacks.** To use SSL, put `SSL = 1` into the config file. If the `URL =` directive is used, make sure to use `https://...` instead of `http://...` there. The ELOG distribution contains a simple self-signed certificate in the `**ssl**` subdirectory. One can replace this certificate and key with a real certitcate to avoid browser pop-up windows warning about the self-signed certificate. `

3.8.2 Using Apache

Another possibility is to use the [Apache](#) web server as a proxy server allowing secure connections. To do so, Apache has to be configured accordingly and a certificate has to be generated. See some [instructions](#) on how to create a certificate, and see *Running elogd under Apache* before on this page on how to run `elogd` under Apache. Once configured correctly, `elogd` can be accessed via `http://your.host` and via `https://your.host` simultaneously.

The redirection statement has to be changed to

```
Redirect permanent /elog https://your.host.domain/elog/
ProxyPass /elog/ http://your.host.domain:8080/
```

and following has to be added to the section `"VirtualHost ....:443` in `/etc/httpd/conf.d/ssl.conf`:

```
# Proxy setup for Elog
<Proxy *>
Order deny,allow
Allow from all
</Proxy>
ProxyPass /elog/ http://host.where.elogd.is.running:8080/
ProxyPassReverse /elog/ http://host.where.elogd.is.running:8080/
```

Then, following URL statement has to be written to `elogd.cfg`:

```
URL = https://your.host.domain/elog
```

There is a more detailed step-by-step instructions at the [contributions section](#).

3.8.3 Using ssh

`elogd` can be accessed through a a SSH tunnel. To do so, open an SSH tunnel like:

```
ssh -L 1234:your.server.name:8080 your.server.name
```

This opens a secure tunnel from your local host, port 1234, to the server host where the `elogd` daemon is running on port 8080. Now you can access `http://localhost:1234` from your browser and reach `elogd` in a secure way.

3.9 How It All Works

For the technically curious:

The concept of **ELOG** is very simple. The logbook functionality is implemented by a single daemon program, `elogd`, which is written in C. It contains an integrated Web server, which does not serve files like standard Web servers, but reads logbook entries from its database and formats them into HTML. Since only forms and tables are used, no Java or Javascript is necessary, which makes the logbook display very fast. The system does not use any images on purpose to reduce the amount of data to be transferred. Since the **ELOG** daemon contains its own `http` server, no additional server like Apache is required.

The "*database*" in which **ELOG** saves its entries is in plain ASCII format. One file is created for each day in the form **YYMMDDa.log** (where YY is the year, MM the month and DD the day). For ELOG versions 1.x.x, the format was **YYMMDD.log**. Messages are separated internally by the string **\$@MID@**. If this string is entered in a message (main body text or attribute), it gets converted automatically in order not to invalidate the database structure.

If attachments are submitted, they are saved as separate files named **YYMMDD_HHMMSS_name** - where in addition to the date the time is specified and **name** is the original file name of the attachment. To copy the database to another computer, only the *.log files and the attachment files need to be copied. To copy for example all files from March 2001, just select them with **0103??a.log** and **0103??_***.

4. ELOG - Syntax of elogd.cfg

Global and individual logbook options for an ELOG server

The configuration file `elogd.cfg` contains entries which define the structure of logbooks and the behaviour of `elogd`. The file has a simple ASCII format. Each logbook is defined by a `[<name>]` section where is the name of the logbook. The `[global]` section is used for settings common to all logbooks. Each line contains a setting name, followed by an equal sign and the value for this setting. Lines starting with ";" are treated as comments.

Here is a simple example, which define two logbooks, "*Linux*" and "*PC*":

```
[global]
SMTP host = mailsend.your.domain

[Linux]
Theme = default
Comment = General linux tips and tricks
Attributes = Author, Type, Category, Subject
Options Type = Routine, Software Installation, Problem Fixed, Configuration, Other
Options Category = General, Hardware, Software, Network, Account, Other
Options Author = Stefan, Linus, unknown
Required Attributes = Author

[PC]
Comment = Database PC installations
Attributes = Location, OS, Owner
Options Location = Building1, Building2
Options OS = Linux, Windows ME, Windows 2000
Required Attributes = Location, Owner
Email All = name@address, othername@otheraddress
Use Mail Subject = Location
```

4.1 Global options

The notation of the following options is such that items enclosed by "<" and ">" should be replaced by a specific string. If a value contains blanks (like a complete sentence), it should **not** be enclosed in quotation marks.

If a setting has a number of possible options, they are shown in the form `option1|option2|...`, meaning that one of the options (without any vertical bar) should be used. The following options are specific to the `[global]` section:

4.1.1 Port = <port>

Specifies the TCP port under which the server is listening. Default is 80. Can be superseded via the '-p' command line flag.

4.1.2 SSL = <0 | 1>

Turn on Secure Socket Layer transport. If SSL is on, one can connect via `https://...` to the `elogd` daemon. If the `URL =` directive is used, make sure to use `https://...` instead of `http://...` there. The ELOG distribution contains a simple self-signed certificate in the `ssl` subdirectory. One can replace this certificate and key with a real certificate to avoid browser pop-up windows warning about the self-signed certificate. The default for this option is `0`.

4.1.3 Interface = <interface>

Specified network interface to listen at. Can be used if several network cards are in a computer, or if one wants to restrict access to the local host only, in which case one can use `127.0.0.1` as the interface.

4.1.4 Resource dir = <directory>

Specifies the root directory for ELOG resources like help files, themes and icons. Can be overwritten with the `-s` flag when starting `elogd`. If not specified, use the directory where the configuration file `elogd.cfg` resides. *Changing this option requires a restart of the `elogd` server.*

4.1.5 Logbook dir = <directory>

Specifies the root directory for logbooks. Can be overwritten with the `-d` flag when starting `elogd`. If not specified, use the directory where the configuration file `elogd.cfg` resides. Each logbook data is stored in a separate directory under this root directory specified by the `Subdir` option. *Changing this option requires a restart of the `elogd` server.* This directory also contains any password file and user HTML file.

4.1.6 Language = <name>

The language setting determines the language of the `elogd` output. Not affected by this setting are the configuration file options and the commands specified with the optional `Menu commands` and `List menu commands`, which have to be specified in English and are translated automatically by `elogd`. The attribute names are unaffected by the language setting and have to be translated manually.

If a language name is given (currently *"german"*, *"french"*, *"spanish"*, *"dutch"*, *"brazilian"* are supported out-of-the-box), the system searches for a file named `eloglang.<name>` containing string translations from English into that language. *If you create a new translation file, please send it back to the author to be included in future distributions.* The online help for `elogd` is contained in the file `eloghelp_XX.html` where *xx* are the first two letters of the language (like *"en"*, *"ge"* and *"fr"*). For new languages, a new file of that type must be created as well.

4.1.7 charset = <name>

Specifies the charset of the pages produced by `elogd`. Can be used to switch to Russian or Asian fonts.

4.1.8 Logbook Tabs = [0|1]

This flag controls the display of *"tabs"* on top of the logbook page which allow to quickly switch between logbooks. Default is `1`.

4.1.9 Main Tab = <string>

If this option is present, an additional first tab is displayed which takes you back to the main logbook selection page. The `string` is used for the contents of the tab.

4.1.10 Main Tab URL = <string>

Normally the main tab brings one back to the logbook selection page. In case one wants to specify a different destination, such as a special web page outside of `elog`, one can use this statement to specify a full URL.

4.1.11 Welcome Title = <html code>

This optional HTML code gets displayed in the title of the logbook selection page. It can contain images via ``. These images must be stored in the resource directory or in the theme directory.

The following line is an example Welcome Title:

```
Welcome title = <p><font size=5 color=white>Welcome to our Elog</font>
```

This displays an image and a text below.

4.1.12 Page title = <string>

The string specified here is used for the title of individual logbook pages. It is also used by most browsers for bookmark names. `<string>` can contain substitutions like `\$<attribute>` where `<attribute>` gets replaced by the attribute string from each message. The option `Page title` in the `[global]` section is used for the logbook selection page.

4.1.13 List page title = <string>

The same for the summary or find result page. This may include substitutions as well, although attribute substitutions make no sense here, since the summary page may contain many messages with different attributes.

4.1.14 Selection page = <file>

When this option is present, a user defined file is displayed instead of the logbook selection page. This file must be stored in the resource directory. Alternatively, an absolute path can be used if the file name starts with a "/" (Unix) or "" or "x:" (Windows). It can be completely customized in order to contain logos etc. As a template, the standard selection page produced by **elogd** can be used.

4.1.15 Guest Selection page = <file>

The same for installations which have a global password file. This means that the logbook selection page is also password protected. It might be however that some logbooks have guest access, in which case guest access to the selection page should be allowed as well (maybe with only a subset of the available logbooks). In that case this options can be used, to show a list of logbooks with guest access.

4.1.16 Protect Selection page = 0 | 1

Normally, one can see the logbook selection page without having to log in. If one wants to require a login for the selection page, this switch can be set to **1**. Default is **0**. It is necessary to put the **Password file = ...** into the *[global]* section of the config file for this to work.

4.1.17 Expand Selection page = 0 | 1

If this option is not present or set to one, the logbook selection page is expanded (all logbooks are shown if groups of logbooks are present). If this option is zero, only the group names are displayed. If one clicks on a group, its logbooks are shown. Using this option set to zero only makes sense if one has a large number of logbooks which would not fit on a single browser window, so collapsing makes sense. Default is **1**.

4.1.18 SMTP host = <host.domain>

This defines the SMTP host needed to send automatic email notifications. The host name you can get from your email program or your local system administrator.

4.1.19 SMTP username = <username>

Some SMTP server require username/passowrd authentication. This option specifies the SMTP user name, while the option **SMTP password** can be created or modified via the **-t** switch when starting **elogd**. This is necessary since the password is encrypted. To set your SMPT password, enter on the command line:

```
elogd -t <your password>
```

4.1.20 SMTP port = <port>

This defines the port under which the SMTP server is listening. The default is 25, but some newer servers use port 587.

4.1.21 Logfile = <file>

This option specifies a filename which logs all login/logout activities and successful user connections for logbooks with user level access. The the **logging level** (see below) is larger than 1, also read and write accesses can be logged.

4.1.22 Logging level = 1 | 2 | 3

Specifies the logging level. The higher this value, the more information is logged. Default is **2**:

- **1**: Log only logins and logouts
- **2**: Log also write accesses
- **3**: Log also read accesses

4.1.23 URL = <http[s]://host.domain[:port]/[subdir/]>

If one of the three cases is true:

- **eelogd** runs with SSL enabled
- **eelogd** runs under a proxy
- The automatic email notifications contains the wrong URL

then the URL under which **eelogd** is running has to be specified manually with this statement. The URL has to contain the port number if not the standard port 80 is used or 433 for SSL, and it has to contain the directory if used under a proxy like

URL	Condition
URL = http://host.domain:8080/	if running on port 8080
URL = https://host.domain/	if SSL is enabled (SSL = 1)
URL = http://host.domain/subdir/	if running under a proxy

This URL is then used for any redirection. For example if one submits a new entry, the URL in the browser reads ...<logbook>/?cmd=Submit&..., containing all the attributes etc. After the submit this page gets redirected to ...<logbook>/<ID>, where <ID> is the ID of the new entry. For the redirection via the HTTP "Location:" statement, an absolute URL is required. Since eelogd cannot figure out the complete URL under which it is running when accessed through an Apache proxy, this statement is necessary to tell eelogd the complete URL.

4.1.24 Relative redirection = 0|1

Under some circumstances, absolute redirection via a complete URL may not work. If you access eelogd through two different ways simultaneously, for example directly and via a tunnel connection, a single absolute URL cannot be used, because one connection starts with **http://**, and the other with **https://**. Another case is when the eelogd server has a dynamic IP address, which changes from time to time. Setting **Relative redirection = 1**, relative redirection is used. This uses the current URL from the browser, whatever it is, and only specifies the last part of the URL. It should be noted however that relative redirections are not allowed in the HTTP standard, but most browsers support it anyhow. Problems have been reported with the Safari browser. So this option should only be used when it is really needed.

```
Usr = <name>
Grp = <name>
```

The user and group to run the eelogd daemon under when started by root.

4.1.25 Resolve host names = 0|1

Resolve remote host names if set to **1**. If set to **0**, which is the default, only IP numbers are stored in any log file. If the **hosts allow/deny** options are used with host names, this setting must be set to **1**. If turned on, the DNS server is contacted on each HTTP request to eelog, which can slow down the server considerably for slow DNS servers.

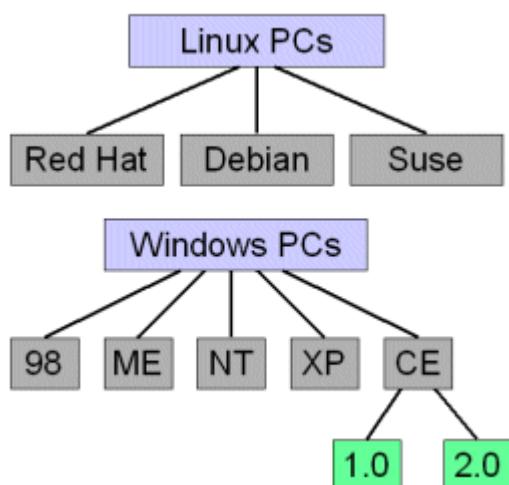
4.2 Groups of logbooks

If installations have very many logbooks, it can be hard to navigate between them. To make things more structured, it is possible to build a hierarchy of logbooks. A logbook group can contain any number of logbooks as well as other logbook groups. The hierarchy is defined with the option

```
Group <group name> = <Logbook1>, <Logbook2>, <other group>
```

in the **[global]** section of the configuration file.

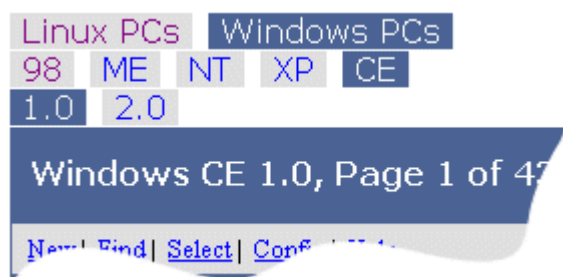
To define following logbook hierarchy:



one would use following statements:

```
[global]
Group Linux PCs = Red Hat, Debian, Mandrake
Group Windows PCs = 98, ME, NT, XP, CE
Group CE = 1.0, 2.0
```

The logbook tabs would then look like this:



Where the selected group or logbook becomes blue. The lower groups/logbooks change according to the selected upper group. Please note that a logbook can be contained in more than one group, but then it should not be the first logbook in those groups. The colors of the tabs and the title bar can be specified in the CSS file.

4.3 Top groups

Sometimes groups of logbooks should be completely separate. Imagine two groups of logbooks, one for the engineering department and one for the administration department. These groups should have different administrators, and the logbook tabs at the top of the screen should not show the logbooks from the other department. Prior to ELOG version 2.4.1, one had to run two elogd servers in parallel, listening under different ports. Since 2.4.1, one can achieve the same behaviour using **Top groups**. The configuration could look like this:

```
Group Linux PCs = Red Hat, Debian, Mandrake
Group Windows PCs = 98, ME, NT, XP, CE
Group CE = 1.0, 2.0

Top group engineering = Linux PCs, Windows PCs
Top group administration = Employees, Purchases

[global engineering]
Password file = engineers.pwd
Admin user = stefan

[global administration]
Password file = admin.pwd
Admin user = bill
```

Note that there can be a `[global]` section for each top level group of logbooks. The rule is that a configuration setting in an individual logbook section overrides a setting in the `[global <top group>]` setting, which by itself overrides a setting in the `[global]` section. This way one can define settings for all top level groups (such as the SMTP host) in the `[global]` section, and define different password files and administrators in the individual top level group sections.

If top groups are used, the root of the elogd server is not accessible any more. Presume that elogd is accessible normally under `http://your.host:8080/`, this URL becomes invalid for top groups, to avoid the case that one group can "see" the logbooks of the other groups. Instead, one has to append the top group name to the URL, such as `http://your.host:8080/engineering` or `http://your.host:8080/administration`. If someone does not know the top group name, one cannot see the list of logbooks there, so the groups become completely independent of each other. If this feature is not wanted, it can be disabled by setting `Show top groups = 1`.

4.4 Individual logbook options

For each logbook, there is a section with the logbook name in square brackets, so that each logbook can have different options. If an option is not present in a logbook section, then the system tries to locate that option in the `[global]` section. Thus if the following options are placed in the `[global]` section, they are defaults for all logbooks. If they are present in the `[global]` and in the logbook section, the logbook option is used.

Here are the available options, by broad categories:

4.5 General options

4.5.1 Data dir = <directory>

This option is obsolete from version 2.2.5 on and should not be used. Use `Subdir = ...` instead.

4.5.2 Subdir = <directory>

Each logbook has a separate directory where the logbook entries are stored, which is controlled by this statement. If the directory does not exist, it is created automatically by the `elogd` program. The subdirectory is relative to the logbook root directory specified with the `Logbook dir = ...` option. So if `Logbook dir = /usr/local/elog/logbooks` and `Subdir = Demo` then the logbook data is stored in `/usr/local/elog/logbooks/Demo`. If the `Logbook dir = ...` option is not specified, then `logbooks` is used. If the subdirectory starts with a "/" ("\" under Windows), then it is used as an absolute path independent of the logbook dir. To see which directories are used, start `elogd` with the "-v" flag.

4.5.3 Comment = <comment>

The comment is displayed on the logbook selection list. The selection list is displayed if more than one logbook is defined on a host and no logbook is explicitly specified in the URL.

4.5.4 Theme = <theme>

A theme determines which layout and colors are used for a logbook, similar to *skins* in other programs. The `theme` option points to a subdirectory under the *"themes"* directory which resides in the resource directory. It contains all files for that theme. The format of these files is described under the *Themes* section.

4.5.5 CSS = <filename>

A given theme can contain several Cascading Style Sheets (CSS). This can be useful if several logbooks use the same images and icons, but different colors. By default, the CSS `elog.css` is used. This statement adds an additional CSS, which can overwrite settings from `elog.css`. If different CSS'es should be used for different output media, this can be accomplished with a comma-separated list in the form `CSS = <file1>&<media1>,<file2>&<media2>`. This will then be translated into separate style sheet statements for the different media. For example a statement `CSS = default.css&screen,print.css&print` will result in the HTML statements:

```
<link rel="stylesheet" type="text/css" href="default.css" media="screen">
<link rel="stylesheet" type="text/css" href="print.css" media="print">
```

4.5.6 Title image = <string>

HTML code for the icon in the upper right corner. By default, following code is used:

```

```

This code can be replaced by `<string>` to display a different icon file, or to display some text. The icon image has to be present in the theme directory, which is usually `<elog root>/themes/default`.

4.5.7 Title image URL = <URL>

The ELOG icon at the right upper corner usually points to the ELOG home page. This URL can be changed to point to a corporate page for example with this option. The icon can be changed by replacing the `elog.gif` icon in the theme directory. This option should only be used if the `Title image` option is not used.

4.5.8 Time format = <string>

This option determines how the date and time of a logbook entry is displayed. The format of the string is the same as the C function `strftime`, so a string of `%A, %B %d, %Y, %H:%M` yields in a display of **Thursday, November 15, 2001, 12:35** for example.

4.5.9 Time format <attribute> = <string>

Same, but just for an individual attribute.

4.5.10 Date format = <string>

This option determines how the date is displayed from attributes which are of type "date". The format of the string is the same as the C function `strftime`, so a string of `%A, %B %d, %Y` yields in a display of **Thursday, November 15, 2001** for example.

4.5.11 Date format <attribute> = <string>

Same, but just for an individual attribute.

4.5.12 Welcome Page = <file>

By default, the list with the last twenty entries of a logbook is displayed when the logbook is selected. This can be overridden with this option, which causes a HTML file to be shown instead of the message list. This file can contain further links for new logbook messages or for logbook queries. Here is a simple example of such a file:

```
<h1>Welcome to the test logbook</h1>
<ul>
  <li><a href="?cmd=new">Enter</a> a new message
  <li><a href="?cmd=find">Search</a> the logbook
</ul>
```

The file must be present in the resource directory. Alternatively, an absolute path can be used if the file name starts with a `"/"` (Unix) or `""` or `"x:"` (Windows).

4.5.13 Start page = <command>

This option can be used to display a different start page. `command` can be either `0?cmd=Last` to display the last message, or any other ELog menu command in the form `?cmd=xxx`. To start with the search page, one uses

```
Start page = ?cmd=Find
```

Please note that if another language than English is selected via the `Language = xxx` option, the commands have to be in that language as well (like `"Start page = 0?cmd=Letzter"` for German).

4.5.14 Submit Page = <file>

This optional page can be displayed when a new message was submitted in a logbook. Here is an example:

```
<h1>You successfully submitted a message</h1>
<a href="?cmd=Back">Back</a> to the logbook<p>
<a href="?cmd=New">Enter</a> another message
```

The file must be present in the logbook directory. Alternatively, an absolute path can be used if the file name starts with a "/" (Unix) or "" or "x:" (Windows).

4.5.15 Message comment = <comment>

This optional comment is displayed on top of the text entry field when submitting a new message. It can contain a sentence like *"Please enter your message here:"*.

4.5.16 Reply comment = <comment>

This optional comment is displayed on top of the text entry field when replying to an exiting entry. It can contain a sentence like *"Please enter your reply here:"*.

4.5.17 Attachment comment = <comment>

This optional comment is displayed on top of the attachment submission section when entering a new message. It can contain a sentence like *"Please upload your attachments here:"*.

4.5.18 Menu commands = <list>

This option specifies the menu commands displayed on top of a single logbook page. For certain installations, it can be useful to disable some commands. Following commands are possible:

- **New** - Enter new logbook entry
- **Edit** - Edit current logbook entry
- **Delete** - Delete current logbook entry
- **Reply** - Submit a reply to current entry
- **Duplicate** - Duplicate the current entry with the possibility to change some values
- **Download** - Download a message in ASCII format
- **Find** - Search entries in logbooks
- **Last day** - Display entries from last day
- **Move to** - Move entry to other logbook
- **Copy to** - Copy entry to other logbook
- **Config** - Edit elogd.cfg (if **no** "Password file" is given)
- **Config** - Modify/Add user accounts (if "Password file" is given)
- **Admin** - Edit elogd.cfg (if "Password file" is given)
- **Login** - Login with user name and password (if "Password file" is given)
- **Import** - Show CSV (comma-separated-values) import page
- **Logout** - Logout current user (if "Password file" is given)
- **Help** - General help

The commands are always in English, independent of the `Language = ...` setting, and are automatically translated into the specified language.

If this option is not present, following default is used:

```
Menu commands = List, New, Edit, Delete, Reply, Duplicate, Find, Config, Help
```

4.5.19 Copy to = <logbook list>

4.5.20 Move to = <logbook list>

The commands **Copy to** and **Move to** make it possible to copy or move a logbook entry from one logbook to another. By default, all logbooks except the current logbook are shown as a possible destination. With the configurations options **Copy to = <logbook list>** and **Move to = <logbook list>** it is possible to specify a list of destination logbooks, separated by commas. This can make sense if only certain logbooks make sense as destinations. The flag **Preserve IDs** can be used to keep the entry ID in the destination logbook.

4.5.21 List Menu commands = <list>

This option specifies the menu commands displayed on top of the listing page. Although all commands from above are possible, only the commands **New**, **Find**, **Select**, **Import**, **Config**, **Admin**, **Change password**, **Logout** and **Help** make sense. The command **Select** can be used to select multiple messages for deletion or for moving to other logbooks. Once the **Select** command is clicked, check boxes appear in front of all entries which let the user select one or more entries. A new menu bar shows up with a **Delete** and optionally a **Copy to ...** and **Move to ...** button, if these commands are present in the **Menu commands** list. Pressing one of these buttons deletes, copies or moves all selected logbook entries.

4.5.22 Guest Menu commands = <list>

This option specifies the menu commands for guest logins. A guest login happens if a password file is used, but someone accesses the logbook for the first time, which means that no username/password is given. In that case the commands from the guest menu are displayed, which usually contain a subset of the normal commands. A typical scenario is a logbook which only has commands to read the logbook on the guest menu, but no commands to write/edit entries. Instead, the **login** command is given in the guest menu, with which one can login as a real user (username and password have to match those from the password file), which then allows full access via the **"Menu commands"** list. A typical example for the menu settings for this scenario are:

```
Menu commands = List, New, Edit, Reply, Duplicate, Find, Config, Logout, Help
Guest menu commands = List, Find, Login, Help
```

Note that the presence of this option opens user access also to the find result or elog listing page, which usually contains some config command. So it is useful to combine the **Guest menu commands** option with the following **Guest List Menu commands** option to restrict the access to the find result page as well.

4.5.23 Guest List Menu commands = <list>

Same as **Guest Menu commands** but for the find result page.

4.5.24 Menu text = <file>

If this option is present, an additional menu row above the message gets displayed with the contents of <file>. This file can contain arbitrary text, images or links. One example would be following text to go back to the listing page and display the next *Routine* entry and all *Routine* entries:

```
<small>
&nbsp;&nbsp;&nbsp;<a href="?cmd=next&type=Routine">Next Routine entry</a>&nbsp;&nbsp;&nbsp;|
&nbsp;&nbsp;&nbsp;<a href="../?Type=Routine">All Routine entries</a>
</small>
```

4.5.25 List Menu text = <file>

The same for the list page.

4.5.26 Filter Menu text = <file>

The same for the filter line in the list page.

4.5.27 Guest Display = <list>

This option specifies which attributes are displayed on guest access. It is possible to display only a subset of all attributes for guest access, but the full list if someone is logged in (using the option "Password file"). The **list** consists of comma separated attributes, including the word *text*, if one wants to display the entry body text for guests.

```
<small>
&nbsp;<a href="?mode=summary">Summary</a>&nbsp;<a href="?mode=full">Full</a>&nbsp;<a href="?mode=threaded">Threaded</a>&nbsp;</small>
```

4.5.28 Top text = <file> | <string>

The text of this option gets displayed at the top of every Elog page. It can be a string or a filename which gets displayed. Might be useful to display company logos etc. If a file is specified, it must be present in the logbook directory. Alternatively, an absolute path can be used if the file name starts with a "/" (Unix) or "" or "x:" (Windows).

4.5.29 Bottom text = <file> | <string>

The text of this option gets displayed at the bottom of every Elog page instead of the little Elog home page link. It can be a string or a file. It can contain for example a link back to the main logbook selection page like:

```
<center><a href="/">Main page</a></center>
```

Or it can contain other useful links. If a file is specified, it must be present in the logbook directory. Alternatively, an absolute path can be used if the file name starts with a "/" (Unix) or "" or "x:" (Windows).

4.5.30 Bottom text login = <file> | <string>

The same as **Bottom text** but for the login page. This allows to display a different text at the bottom of the login page. It can also be used to execute some JavaScript.

4.5.31 Help URL = <URL>

This URL is used for the Help button. By default, the file **eloghelp_xx.html** is returned with the contents of the help page. Edit this file directly to add site-specific help for all logbooks. Alternatively, use the **Help URL** option to specify different help pages for different logbooks. It can point to a site-specific help page via **http://...** or to a local file like **file://c:/tmp/config.html**, or to the name of an HTML file which must be present in the resource directory.

4.5.32 Message Width = <number>

This value sets the number of characters per line of the main message entry field. The default value is 76 (78 for replies), and can be increased for installations which need a larger window size (like pasting log files etc.). If both **Message Width** and **Message Height** are not given, some JavaScript code is used which automatically resizes the message window dynamically to fit optimally into the browser window.

4.5.33 Message Height = <number>

This value sets the number of lines of the main message entry field. The default value is 20, and can be changed for installations which need a different window size. If both **Message Width** and **Message Height** are not given, some JavaScript code is used which automatically resizes the message window dynamically to fit optimally into the browser window.

4.5.34 Admin textarea = <cols>,<rows>

This defines the textarea size for the admin page. Default is **80,40**.

4.5.35 Display mode = [full|summary|threaded]

Default mode for search display. On the find entry form, the checkboxes are set accordingly. The "Last xxx" page uses this setting directly.

4.5.36 Entries per page = <number>

Number of logbook entries displayed per page in a search result. The default is 20.

4.5.37 Restrict edit time = <hours>

If this option is set, a new message can only be edited a certain number of hours after its creation. This can be useful if one wants to ensure that old entries cannot be modified. Hours can also be fractional, like 0.5 for 30 min.

4.5.38 Admin restrict edit time = <hours>

Same option for admin users. This can be useful if normal users are not allowed to change entries after "restrict edit time", but an admin user should be allowed to do so. Setting this to zero disables any restriction for admin users and they can edit entries forever.

4.5.39 Max content length = <bytes>

This option restricts the size of attachments. When very large (>100MB) attachments are uploaded, the elogd server can be busy with this upload for a longer time and not respond to other requests during that time. To avoid this, the maximum size of attachments can be restricted. The server will then refuse to accept larger attachments. The default is 10485760 (= 10 MB). This option has to be placed into the [global] section and the elogd server has to be restarted after a change.

4.5.40 Fonts = <list>

List of fonts (comma separated) to be shown in the font drop-down box of the entry edit form. Default is

Fonts = Arial, Comic Sans MS, Courier New, Tahoma, Times New Roman, Verdana

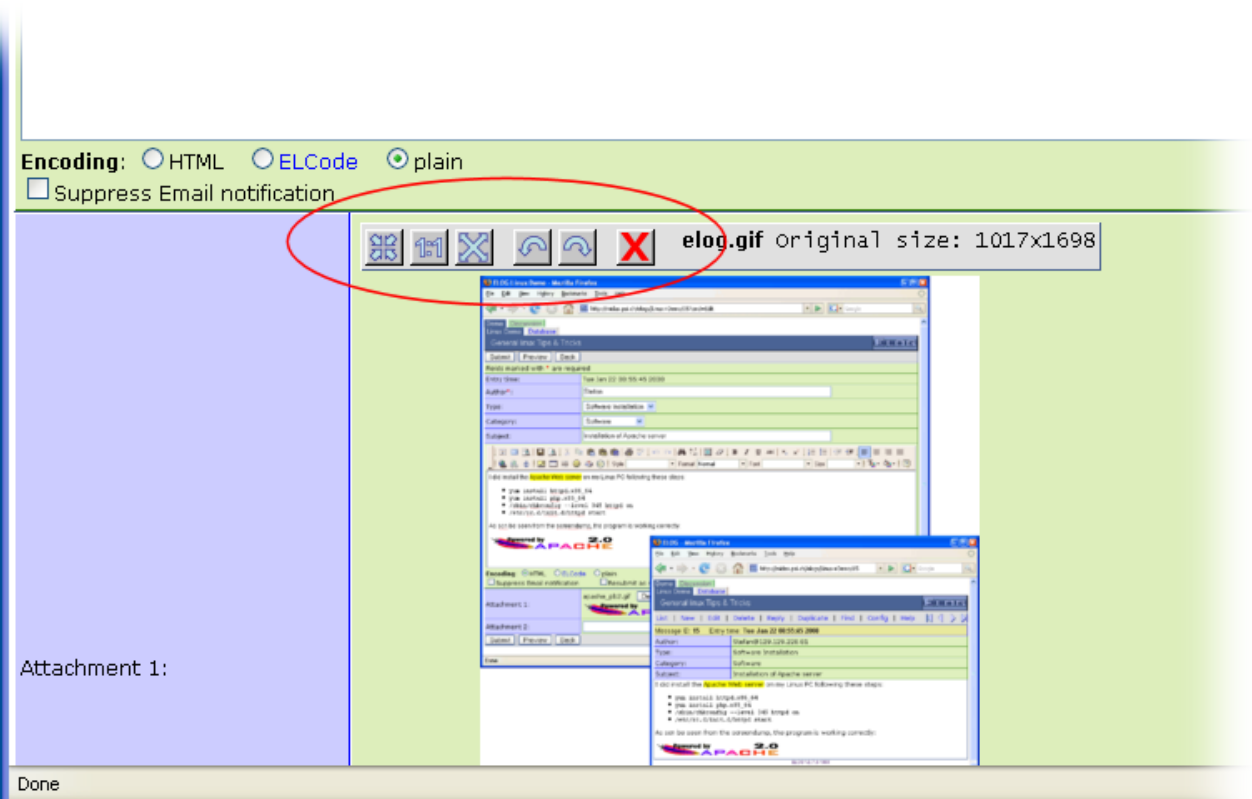
On Unix systems some of these fonts might not be installed, in which case they can be replaced by others like **Serif**, **Sans-serif**, **Helvetica**.

4.5.41 All display limit = <n>

If a logbook contains many entries, the list gets divided into pages, with some page navigation for the next, previous, a specific page and all pages. If the logbook contains a large number of entries (>500), the display of all these entries can take very long and might slow down the elogd server, especially if the entries are not displayed in "summary" mode but in "full" mode. Therefore the "All" link should not be used in the page navigation for large logbooks. The number of entries from when on the "All" link gets hidden can be specified with this number, the default value is **500**.

4.5.42 Thumbnail size = <size>

This option determines the default thumbnail size. To make the automatic generation of thumbnails working, the ImageMagick package has to be installed. Refer to the [admin guide](#) for installation instructions. The thumbnail size **size** gets passed to the **-thumbnail** option of the conversion. A value of **300** converts all pictures to thumbnails 300 pixels wide. A value of **300>** converts all pictures to thumbnails 300 pixels wide if they are larger than 300 pixels initially, and leaves them untouched if they are smaller. A value of **10%** converts all pictures to 10% of their original size. If the thumbnail size option is missing, the thumbnails will be created with the original image size, and can then be resized and rotated interactively with the image manipulation buttons:



Setting **Thumbnail size = 0** turns off the thumbnail creation.

4.5.43 Thumbnail options = <options>

With this option one can pass additional parameters to the ImageMagick package. They are passed 1:1 to the **convert** program. Commonly used is the **-density** option to increase the image quality when converting from PDF or EPS files.

4.6 Attributes

4.6.1 Attributes = <list>

Define a number of attributes for the logbook, separated by commas. A maximum of 100 attributes can be defined. Typical values are "Author", "Subject" or "Type". Following values are not allowed:

- Text
- Date
- Encoding
- Reply to
- In reply to
- Locked by
- Attachment
- Path

since these are used internally by elog.

4.6.2 Options <attribute> = <list>

Usually, an text field is used for an attribute, where the user can fill in text of up to 100 characters. If instead a drop-down box with preset items is better for a given attribute, these items can be defined with this statement. Up to 100 items can be defined, separated by commas. To add an option including a comma, enclose it in quotations marks like

```
Options town = San Francisco, "Paris, Texas", "Paris, France"
```

4.6.3 Extendable options = <list>

When using the **Options <attribute>** to specify a list of possible options, this list is fixed. Sometimes it is desirable to extend the list when a new entry in a logbook is made and a certain option is missing on the list. By adding the attribute name to the **Extendable options** list, a button appears next to the attribute in the message entry form which lets you add new options to the list. The elogd.cfg configuration file is then automatically updated. When a new logbook entry gets made, the new option automatically appears in the drop-down box for that attribute.

4.6.4 ROptions <attribute> = <list>

Same as **Options** above, but using radio buttons instead of a drop-down box.

4.6.5 MOptions <attribute> = <list>

This list allows for "*Multiple Options*", meaning that an attribute can have several values simultaneously. When entering an entry with MOptions, each value from the list is represented by a checkbox. Unlike with normal options, multiple checkboxes can be checked for an entry. The attribute value then becomes

```
<value1> | <value2> | ...
```

In the "*find*" page only one of these values can be specified, which is then treated as a substring in the search filter.

4.6.6 IOptions <attribute> = <list>

This list specifies a set of icons for an attribute. Some icons are contained in the *themes/default/icons* directory which can be used here like

```
Attributes = Author, Icon, Subject...
IOptions Icon = icon1.gif, icon2.gif, icon3.gif, ...
```

New icons are welcome and should be sent back to the author to be incorporated in the next version.

4.6.7 Comment <attribute> = <comment>

Optional comment which is displayed below the attribute name in the entry form. Can be used to explain the attribute somehow.

4.6.8 Tooltip <attribute> = <comment>

Same as **Comment <attribute>**, except that the comment gets displayed as a tooltip (tiny pup-up window) when the user moves the mouse cursor over the attribute name in the entry form.

4.6.9 Tooltip <attribute> <attribute option> = <comment>

Same as **Tooltip <attribute>**, but for option values of a **MOptions** attribute. Using this option, a different tooltip can be shown above each check box of an optional value for an attribute. Please note that attributes or options with spaces should **not** be enclosed with quotes.

4.6.10 Icon comment <icon> = <comment>

Icons may contain a comment, which is then used in email notifications instead of the icon file name. One has to add a separate icon comment for each icon file.

4.6.11 Options <attribute> = boolean

If an attribute is marked "*boolean*" this way, a checkbox is displayed for this attribute.

4.6.12 Preset <attribute> = <string>

This option uses a preset string for an attribute. The string can contain substitutions like the ones described under the "*Subst <attribute>*" command. One possible application is to use the login name for the author field like:

```
Preset Author = $long_name
```

If the attribute should be locked at the Web submission, use the "*Locked Attributes = ...*" option. If a preset value is given for an attribute which has an options list, the preset value is selected in the drop down box by default.

4.6.13 Preset text = <string> or <file>

This preset value is used for the main body text. It can be a string or a file, which must be present in the logbook directory. Alternatively, an absolute path can be used if the file name starts with a "/" (Unix) or "\" or "x:" (Windows).

4.6.14 Preset on edit <attribute> = <string>

Same as **Preset <attribute>**, but evaluated when editing existing entries.

4.6.15 Preset on reply <attribute> = <string>

Same as **Preset <attribute>**, but evaluated for replies.

4.6.16 Preset on first reply <attribute> = <string>

While **Preset on reply <attribute>**, is evaluated for any replies, this one is only executed for the first reply to an entry. It can be useful for example to do something like this:

```
Preset on first reply Subject = Re: $Subject
```

So the "Re:" only gets added once, and you don't get long chains of "Re: Re: Re:".

4.6.17 Preset on duplicate <attribute> = <string>

Same as **Preset <attribute>**, but evaluated for duplicated entries.

4.6.18 Locked Attributes = <list>

The attributes specified here cannot be modified when a new entry is submitted. This makes only sense for preset attributes.

4.6.19 Fixed Attributes Edit = <list>

The attributes specified here cannot be modified when an existing entry is modified via the **Edit** button. This feature can be useful to preserve the original author of the message, when using the **Preset Author = \$long_name** option as described above.

4.6.20 Fixed Attributes Reply = <list>

The attributes specified here cannot be modified when an existing entry is replied on via the **Reply** button. This feature can be useful to preserve the original subject of a message for example.

4.6.21 Required Attributes = <list>

The attributes specified here are required when a new entry is submitted. The attribute names are marked with * on the entry form.

4.6.22 Show Attributes = <list>

Attributes present in this list are shown in the single entry page. Omitting attributes can make sense for attributes which are automatically derived from other attributes via the **Change <attribute>** command.

4.6.23 Show Attributes Edit = <list>

The same as **Show Attributes** , but for the entry form.

4.6.24 Propagate Attributes = <list>

With this option, changed in an attribute are automatically propagated to all entries of a thread. This can be useful if one has an attribute "problem status" for example with the options "open", "under investigation", "fixed". A thread related to a specific problem can then have several replies. If the problem gets fixed, a new reply can be made with the attribute "problem status" being "fixed", and then the propagation causes all entries of this thread to become "fixed".

4.6.25 Page title = <string>

The string specified here is used for the title of the web page. It is also used by most browsers for bookmark names. The string can contain substitutions as described under the "*Subst <attribute>*" option.

4.6.26 Edit Page title = <string>

The string specified here is used for the title of the entry form. It is also used by most browsers for bookmark names. The string can contain substitutions as described under the "*Subst <attribute>*" option.

4.6.27 List display = <list>

Specified the display and order of items in a message listing page or a search result page. In addition to all attributes, following items can be specified:

- **ID** for the entry ID
- **Date** for the entry date/time
- **Edit** to display a column with an edit icon to directly edit and entry
- **Delete** to display a column with a delete icon to directly delete and entry

The restriction to certain attributes can be helpful if many attributes are defined in a logbook, which usually makes the table too big to fit in the browser. The default is\

```
List display = ID, Date, <all attributes>
```

Which displays the message number, date, and all attributes. The display of the message body is controlled by the **Display mode** and **Summary lines** options. If a search goes over "all logbooks", an additional columns with the logbook name of each entry is added in front.

4.6.28 Guest List display = <list>

Same as **List display** , but for guest access (user level access with password, but not logged in). Please see also **Guest display** . In addition to **List display** , one can optionally specify **Text** as an attribute here. Without that attribute, the summary text of the entry body is not shown. This makes it possible to show the text for registered users and hide it for guest access.

4.6.29 Link display = <list>

Normally, each column in the display list contains a link to the individual entry. If this is not desired, the list of attributes with links can be restricted to only a subset with this option.

4.6.30 Thread display = <string>

Optional way to specify the line contents in the threaded search result. Following substitutions are possible:

- **\$<attribute>****: The value of the attribute
- **\$logbook****: The name of the current logbook
- **\$entry time**: *The message date and time, formatted via "Time format"*
- **\$message id****: The message ID

A typical example would be

```
Thread display = $subject, posted by $author on $entry time
```

4.6.31 Thread icon = <attribute>

If a logbook uses some icons for an attribute, these icons can be displayed in the search result page instead of the default icons contained in the themes directory.

4.6.32 RSS Title = <string>

ELOG supports so-called *RSS feeds*. Once can subscribe to new logbook entries with RSS readers such as Mozilla Firefox. Once new entries are submitted to the logbook, they become visible in the subscription. By default, all attributes of the last 15 logbook entries are used as the RSS title. With this option one can change this behaviour. Following substitutions are possible:

- **\$<attribute>**: The value of the attribute
- **\$logbook**: The name of the current logbook
- **\$entry time**: The message date and time, formatted via "*Time format*"
- **\$message id**: The message ID

A typical example would be\

```
RSS Title = $subject, posted by $author on $entry time
```

4.6.33 RSS Entries = <n>

Number of entries to be shown in the RSS feed. Default is 15.

4.6.34 Subst <attribute> = <string>

When submitting logbook entries, attribute values can be substituted by some text. This text can contain arbitrary fixed text and following values:

- **\$<attribute>**: The entered value of the attribute itself
- **\$host**: The host name where **elogd** is running
- **\$remote_host**: The host name of the host from which the entry was submitted
- **\$short_name**: The login name (if password file is present)
- **\$long_name**: The full name from the password file for the current user
- **\$user_email**: The email address from the password file for the current user
- **\$logbook**: The name of the current logbook
- **\$date**: The current date, formatted via "*Date format*"
- **\$utcdatetime**: The current UTC date (GMT) and time, formatted via "*Date format*"
- **\$version**: The version of the ELOG server in the form x.y.z
- **\$revision**: The Subversion revision of the ELOG server as an integer number
- **\$shell(<command>)**: <command> gets passed to the operating system shell and the result is taken for substitution.

Following example use this feature to add the remote host name to the author:\

```
Subst Author = $author from $remote_host
```

Following example substitutes an attribute with the contents of a file:

```
Subst Info = $shell(cat /tmp/filename)      (Unix)
Subst Info = $shell(type c:\tmp\filename)    (Windows)
```

A special option are automatically generated tags, which are automatically incremented for each new message. This is achieved by putting #'s into the substitution string, which is used as a placeholder for the incrementing index. Each "#" stands for one digit, thus the statement

```
Subst Number = XYZ-####
```

results in automatically created attributes "*Number*" of the form

```
XYZ-00001
XYZ-00002
XYZ-00003
```

and so on. In addition to the #'s one may specify format specifiers which are passed to the [strftime](#) function. This allows to create tags wich contain the current year, month and so on. Once the date part of the attribute changes, the index restarts from one. The statement

```
Subst Number = XYZ-%Y-%b-###
```

results in automatically created attributes "*Number*" of the form

```
XYZ-2005-Oct-001
XYZ-2005-Oct-002
XYZ-2005-Oct-003
```

and

```
XYZ-2005-Nov-001
XYZ-2005-Nov-002
```

on the next month.

4.6.35 Remove on reply = <list>

This option clears one or more (separated by commata) attribute values from a logbook entry when creating a reply to that entry. This can make sense for example for the author, since the author of a reply can be different from the original author.

4.6.36 Quote on reply = 0 | 1

This flag controls if the original text is quoted in a reply. Default is 1

4.6.37 Reply string = <string>

String used to mark original message lines. Default is "> " . Can be empty string ("") if no message marking is desired.

4.6.38 Subst on reply <attribute > = <string>

Substitution of attributes for replies. This option can be used to replace the current subject with a "Re: <old subject>":\

```
Subst on reply subject = Re: $subject
```

Note that this option works only for the first reply. So a reply-to-a-reply would still have **Re: <old subject>** and not **Re: Re: <old subject>**. If you want the substitution for all replies, please use **Preset on reply** instead.

4.6.39 Subst on edit <attribute > = <string>

Substitution of attributes for edited messages. This option can be used to replace the author by the current author for example:\

```
Subst on edit author = $full_name
```

4.6.40 Quick filter = <list>

Specifies list of comma separated attributes for which a drop-down filter is displayed in the search result page. By selecting a value from that drop-down box, only entries with that value are displayed. In addition to all attributes defined in the **Attributes =** list, the attribute **Date** and the option **Subtext** can be listed here. Using the **Date** filter, the last day, week, month and so on can be displayed. The **Subtext** filter works on the entry body text.

4.6.41 Last default = <n>

Some logbooks are very big and searching through all entries with a quick filter can be time consuming. This option sets a default value for the **Date** quick filter, so that by default only the <n> last days are displayed. <n> has to match one of the entries of the data quick filter options, which are 1, 3, 7, 31, 92, 182, 364.

4.6.42 Format <attribute> = <flags>,<css_class_name>,<css_class_value>,<width>,<size>

Optional formatting parameters for attributes. Following items can be defined in the comma-separated list:

Values used for single message display page:

- **<flags>** Sum of following flags:
 - **1**: Display attribute in same line as previous attribute
 - **2**: Display radio buttons or check boxes in separate lines (if applicable)
- **<css_class_name>,<css_class_value>** *Cascading Style Sheet class names used for cells containing attribute name or value, respectively. The classes must be defined in the style sheet file (usually themes/default/default.css).*

Values used for new message entry form:

- **<width>** Width of the text entry field in characters
- **<size>** Maximum number of characters allowed.

Default is "0, attribname, attribvalue, 80, 500". Trailing parameters can be omitted, so specifying for example only the flags is possible.

4.6.43 Type <attribute> = date | datetime | numeric | userlist | useremail | muserlist | museremail

A normal attribute can contain strings of any type. With this option, attributes can be forced to be numeric or to be a date/time, or to consist of a list of all users from the password file. When new logbook entries are made, numeric attributes are checked to contain only digits. Note that JavaScript has to be enabled to do this.

Attributes of type **date** are treated as a date. Their format for display can be controlled by the **Date format** option. Upon entry, drop-down boxes are displayed which let the user select the day, month and year. Alternatively, a pop-up date picker using a calendar can be displayed if JavaScript is enabled. Date attributes are saved internally as seconds since 1.1.1970, and can therefore be sorted properly by clicking on the header of a logbook entry list. On the find page, dates can be searched for via a start and end date. If date attributes are used in a quick filter (see above), a drop-down quick filter box is displayed which lets the user select "last day", "last week", "next week", and so on. The **datetime** type combines a date and time in HH:MM. The output of this combination is controlled by the **Time format** option.

If the attribute type is **userlist**, a drop-down box is displayed which contains all user names from the current password file. This can be useful for example in a bug tracking system, where a new entry gets assigned to an individual. The type **useremail** is similar, just a list of email addresses of all registered users. This can be used to send email notification to assigned people by using this attribute in an **Email all = <attribute>** statement. The type **muserlist** and **museremail** are the same that **userlist** and **useremail**, except that several user names or user emails can be selected at once using check boxes.

4.6.44 Style <attribute> <value> = <style>

Optional formatting of logbok entries in list mode. For some logbooks it might be useful to display different entries in a different color for example. To achieve this, a CSS style sheet can be attached to an entry based on the value of an attribute. If you have an attribute called **importance** and you want to highlight all entries where **importance** is **severe** for example, you can specify following style:

```
Style importance severe = background-color:red
```

For possible formattings, please refer to some CSS documentation. You can change the colors, font styles and sizes. The style is then valid for the whole row of that entry.

For empty attributes one can specify "", such as

```
Style importance "" = background-color:red
```

4.6.45 Cell Style <attribute> <value> = <style>

Same as above, but only for a specific cell containing <attribute>. Following options

```
Attributes = Author, Status
Options Status = Fixed, Under Process, Not Fixed
Cell Style Status Fixed = background-color:green
Cell Style Status Not Fixed = background-color:red
Cell Style Status Under Process = background-color:yellow
```

for example produce following listing:

Demo					
Demo, Page 1 of 1					ELOG
New Find Select Import Config Last day Help					
Full Summary Threaded					3 Entries
ID	Date	Author	Status	Text	
1	4/15/2010 11:26:13 AM	Stefan	Fixed	Problem has been fixed today.	
2	4/15/2010 11:26:20 AM	Martin	Under Process	I'm currently working on it.	
3	4/15/2010 11:26:24 AM	Vera	Not Fixed	New problem showed up.	

ELOG v2.7.8-2294

4.6.46 Change <attribute> = <string>

Instead of substituting an attribute, the original attribute can be kept and just the output formatting can be changed. This can be very handy for constructing HTML links out of attributes. Presume that a company has a telephone book reachable under

```
http://any.company.com/telbook.cgi?search=<name>
```

where <name> has to be replaced by a search string. Now one can construct an automatic telephonebook lookup with following options:\

```
Attributes = Name, Telephone, ...
Change Telephone = <a href="http://any.company.com/telbook.cgi?search=$Name">$Name's telephone number</a>
```

The attribute **Telephone** is now automatically constructed from the attribute **Name** and consists of a link to the company's telephonebook. The advantage of this system is if the URL of the telephonebook changes one day, only one statement in the config file has to be changed, while otherways (like with the **Subst Telephone = ...** option) all entries would have to be changed manually.

4.6.47 List Change <attribute> = <string>

Same option for the list display.

4.6.48 Execute new | edit | delete = <command>

It is possible to execute a shell command on the server side after a new message has been submitted, edited or deleted. This feature has been used in the past for SMS notifications over a telephone system and for synchronization of the ELOG database with an external SQL database. The `<command>` can contain substitutions similar to the `Subst` command. In addition the list of all attachments can be referred to via `$<attachments>`. The text body of the entry can be referred to with `$text`. It should be noted that only the first 1500 characters of the text can be used, in order not to exceed the limits of the shell. Following (Unix) command writes a notification into some file:

```
Execute new = echo "New message wiht ID $message id of type $type from $long_name on $remote_host" >> /tmp/elog.log
```

It should be noted that this feature can impose a security problem. If someone can edit the `elogd.cfg` through the `Config` command of `elogd`, that person can put malicious code into `elogd.cfg` and execute it. This is even more severe if `elogd` runs with root privileges. To avoid such problems, the execute facility is disabled in `elogd` by default and has to be enabled explicitly with the `"-x"` command line flag. The administrator has to ensure then of course that only trusted people can edit `elogd.cfg`.

4.6.49 Last submission = <string>

This option determines what gets displayed on the logbook selection page in the *Last submission* colum. The default string is `$entry time by $author`. If a logbook does not contain an `author` attribute, another string can be chosen.

4.6.50 ID display = <string>

This option determines the display of the entry ID. In some applications, the entry ID can be used as a tag, containing more than just the ID number. For example

```
ID display = TAG-$message id
```

would display the entry ID as "TAG-1", "TAG-2", ... and so on.

4.6.51 Prepend on reply = <string>

With this option a string can be placed on top of a reply. Using string substitution, this can be useful for adding the author and the date of a reply, like

```
Prepend on reply = Added $date by $long_name\n\n
```

where `"\n"` causes a line break.

4.6.52 Append on reply = <string>

Same as before, but gets added after the previous entry.

4.6.53 Prepend on edit = <string> **

4.6.54 Append on edit = <string>

Same as before, but for editing entries.

4.6.55 Sort Attributes = <list>

For the list display, the entries are normally sorted by their ID. Alternatively, one can specify one or more (separated by commata) attributes, which are used for sorting. The first attribute in the list has the highest priority. Only if two entries have the same value in the first sort attribute, they are sorted according to the second sort attribute and so on. To the list of attributes one can add **ID**, **Date** and **logbook**, although **ID** makes only sense together with other attributes, since it is sorted as the primary key anyhow.

4.7 Conditional attributes

When entering data into a elog form, it might be helpful to change the options of the attributes depending on the value of other attributes. Let's assume you have a logbook containing entries for different computers with different operating systems. Your `elogd.cfg` file starts like that:

```
Attributes = PC Name, Operating System, Version
Options Operating System = Linux, Windows
```

For the operating system version, you would like a list, but this list has to be different for Linux and Windows. This can be achieved with *conditional attributes*. Simply write following configuration:

```
Attributes = PC Name, Operating System, Version
Options Operating System = Linux{1}, Windows{2}
{1} Options Version = 2.2, 2.4, 2.6
{2} Options Version = ME, 2k, NT, XP
```

If you enter a new entry into that logbook, the drop-down list for **Version** changes automatically depending on the **Operating System**. Note that you have to enable Java Script for this to work. Without Java Script, a separate button appears in the line of the Operating System which has to be pressed to make the Version list change.

The number {1} and {2} in the configuration file are called *conditions*. Depending on these conditions, certain other lines can be activated. So if the Operating System *Linux* is selected, condition {1} is true, which selects the line starting with {1} to select the options *2.2, 2.4, 2.6*.

This technique offers various other possibilities, since any configuration option can be made conditional by adding a `{<n>}` in front of that line where `<n>` is an arbitrary number. One often used possibility is the definition of forms. Depending on an attribute, the configuration option **Preset text** = ... can be used to copy some pre-defined forms into the message body, which can then be filled out. Consider following example:

```
Attributes = Author, Type
Options Type = Network check{1}, System check{2}

{1} Preset text = network.txt
{2} Preset text = system.txt
```

This causes two text files *network.txt* and *system.txt* to be copied into the message body when a new entry is made. The file *network.txt* could look like:

```
Routers checked: [ ]
DHCP checked:   [ ]
Comment: ...
```

This works like a pre-defined form, the user puts X's between the "[]" when that item has been checked. Other possibilities are pre-defined shift sheets in environments where elog is used as a shift logbook. The shift sheet could contain the names of the shift crew, some check-list for standard tasks etc.

Another use of conditional attributes is in conjunction with the option **Message comment**. Depending on some attribute values, different message comments can be displayed to tell the user what to enter exactly in the message body for that attribute value.

4.7.1 Show Attributes Edit = <list>

When using conditional attributes, it might be necessary to omit certain attributes under certain conditions, to make the input mask shorter and maybe change the order of the attributes. With this option, a subset of all attributes can be specified which get displayed on the single entry page in the same order as they are specified here. This option mainly makes sense when used with conditions, such as:

```
Attributes = PC Name, Operating System, Version, Distribution
Options Operating System = Linux{1}, Windows{2}
{1} Show Attributes Edit = Operating System, Distribution, PC Name
{2} Show Attributes Edit = Operating System, PC Name, Version
```

The above statements cause the attribute **Version** to be only visible when "Windows" is selected, and **Distribution** to be only visible when "Linux" is selected. If "Windows" is selected, the PC name is shown before the version.

4.8 Multiple conditions

It is possible to define conditions in more than one options list. The only requirement is that conditions are unique, meaning that a condition in one option list cannot be used in another list. This can easily be avoided by using numbers for one condition and letters for the other condition, like in the following example:

```
Attributes = PC Name, Operating System, Version, Location, Floor
Options Operating System = Linux{1}, Windows{2}
Options Location = Main Building{a}, New Building{b}, Old Building{c}
{1} Options Version = 2.2, 2.4, 2.6
{2} Options Version = ME, 2k, NT, XP
{a} Options Floor = Ground, First, Second
{b,c} Options Floor = Ground, First
```

It is possible to specify an OR of several conditions like in the case {b,c}. This is also possible over several conditions, like {1,a} would mean *"The PC has Linux or is in the Main Building"*. To specify a AND between conditions, a "&" is used. The condition

```
{1&a} ...
```

specifies for example the condition "Linux AND Main Building". If several lines with condition combinations are true, the upper one is used.

4.9 Conditions in the list display

Conditional attributes are usually only used for change items in the entry form. It might however be desirable to have conditional attributes also working in the list display (the page where several entries are shown on a single page). The value of one attribute can then for example change which other attributes gets displayed via the **list display** option. To enable the evaluation of conditional attributes for the list display, one uses the option

4.9.1 List conditions = 1

It should be noted that this option can cause a significant performance degradation if many conditional attributes are defined, so it should only be turned on when it is really needed.

4.10 Access control

Note: Starting with version 2.9.0, the password level access using the options *Read password*, *Write password* and *Admin password* is not supported any more. Please use the user level access as described below.

4.11 Password file

Access control is done on a user level with a password file. When a user logs in, a session ID is created and placed as a "cookie" in the browser. Using this cookie, the user can work on the logbook until the cookie expires. For this it is necessary that cookies are enabled in the browser.

Following options can be used to control the behavior:

4.11.1 Password file = <file>

4.11.2 Login expiration = <hours>

4.11.3 Admin user = <user list>

4.11.4 Login user = <user list>

This file contains user names and passwords in XML format, such as

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- created by MXML on Tue Nov 07 08:15:51 2006 -->
<list>
  <user>
    <name>stefan</name>
```

```

<password encoding="SHA256">Ebx/a.9tFFQ/iUW3mU8GbnPpCVk74jFt56CmiJXVwdm</password>
<full_name>Stefan Ritt</full_name>
<last_logout>Tue Oct 17 12:59:47 2006</last_logout>
<last_activity>Tue Nov 07 08:15:51 2006</last_activity>
<email>stefan.ritt@psi.ch</email>
<email_notify>
  <logbook>demo</logbook>
</email_notify>
</user>
<user>
  <name>midas</name>
  <password encoding="SHA256">t56CmiJXVwdmEbx/a.9tFFQ/iUW3mU8GbnPpCVk74jF</password>
  <full_name>Midas User</full_name>
  <last_logout>0</last_logout>
  <last_activity>0</last_activity>
  <email>midas@psi.ch</email>
  <email_notify>
    <logbook>demo</logbook>
  </email_notify>
</user>
</list>

```

The passwords are encoded. New users can either be created by hitting **Register as new user** on the login page if **Self register = 1** in the configuration file, or by the admin user in the **Config** page by pressing **New user**. The password file resides in the same directory as the logbooks. When a user is logged in, the entry for this user can be modified via the **Config** command.

To start a new password file, follow these steps:

- Specify a password file name with **Password file = <file>** in the configuration file
- Connect to the logbook. You will be presented with the new user page. Enter the user login name, full name, email and password, then click on the "Save" button.
- Add **Admin user = <user>** into the configuration file, using your login name from above
- If you now enter the "Config" page, you can add other users
- Remove the self registration option if you like

The presence of a password file requires all users to "*log in*" using their name and password, except when a guest login is allowed via the "**Guest menu commands**" option. An additional advantage of this method is that the user name can be used as an attribute value for creating logbook entries. For example, the following line could be added to the configuration file to fill in the *Author* and the *Email* attributes with the current user name and email:

```
Attributes = Author, Email, ... Subst Author = $long_name from $remote_host Subst Email = $user_email
```

Thus the author name is not user-input anymore, ensuring the entry always contains the actual user name. For a full listing of substitutions, see the "**S ubst <attrib>**" option.

The user name and password are stored as cookies on the user side. The expiration is controlled by the **Remember me** checkbox during the login. If unchecked, the cookies expire after the current browser session. If checked, they expire after 31 days by default, which can be changed with the **Login expiration** option, giving the expiration time in hours. Setting this to 24 for example, makes the password expire after one day. If persistent cookies are not desired, the **Login expiration** option can be set to zero, in which case the **Remember me** checkbox is not displayed.

The **Admin user = <user list>** is a list of one or more user names, which have admin rights. They see a button **Change elogd.cfg** on the config page by which they can edit `elogd.cfg` through the web. They can also modify other users on the **Config** page, change their passwords or remove them. In addition, the admin user(s) can delete or edit entries from other users if **Restrict edit = 1**.

The **Login user = <user list>** is a list of users who can log in to a specific logbook. This option can be used with a global password file. If a **Password file** is present under the **[global]** section, the registered users in that password file can log in to all logbooks. It might be required that only certain users can log in to certain logbooks. This can be achieved with the **Login user** option, places in each individual logbook section in the configuration file. Only those users listed in this statement can log in to the logbook where the statement is defined. This method has the advantage over the option of defining individual password files for individual logbooks that only one central password file exists. So if a user changes her/his password, this becomes then valid for all logbooks. If there would be individual logbook password files, one would have to change the password in all logbooks individually.

4.11.5 Self register = 0|1|2|3|4

With this option it is possible for new users to self-register an user account. At the login page, a link is displayed "**Register as a new user**" which leads the user to a configuration page where one can enter the account name, full name and email address. A flag allows for automatic email notification on new entries on the logbook. These settings can later be changed with the **Config** menu command.

Setting this option to **0** disables self registration. With option **1**, users can silently register, while setting it to **2** causes elogd to send an email notification to the admin user(s). The option **3** is used to *only* send an email notification to the admin users(s), which then can validate the account and commit it by hitting the URL given in the email notification. Setting this to **4** causes an email notification to be sent to the user, which then can validate the account herself/himself proving to have a working email account.

4.11.6 Allow password change = 0|1

Enables or disabled the ability for users to change their password. If disabled, the "*Forgot password?*" link in the login page is omitted as well. The admin user(s) can always change passwords.

4.11.7 Allow <command> = <user list>

Commands can be restricted to certain login names (separated by commas). For each command in the list defined with the "*Menu commands*" option, a list of user names can be specified, which are allowed to execute that command. If the allow option is not present, all users may execute that command by default.

4.11.8 Deny <command> = <user list>

Used to deny a certain command to a list of users. This can be used to deny a guest user to enter new messages or modify a message.

4.11.9 Hosts allow = <list>

4.11.10 Hosts deny = <list>

These two settings can be used to restrict the access to the logbook to certain computers. It is similar to the UNIX *hosts.allow* and *hosts.deny* files. The list can consist of individual host names or IP numbers, subnet masks like **123.213.** (note the trailing **.**) or **.mit.edu**, or the word **All**. The following rules are applied:

- Access will be granted when a host matches a pattern in "*hosts allow*".
- Otherwise, access will be denied when a host matches a pattern in "*hosts deny*".
- Otherwise, access will be granted.

These rules are applied *before* any password is checked. To debug problems, start **elogd** with the "-v" flag, in which case the rule checking is printed on the screen.

The global option **Logfile = <filename>** can be specified to log all user login/logout activities plus all successful user connections.

If any of the password statements are in the **[global]** area of the configuration files, they are used for all logbooks. If one logs in at one logbook, access is automatically granted to all logbooks. If the password statements are in the individual logbook sections, one has to log in to each logbook separately.

4.12 Kerberos authentication

Starting from version 2.9.0, site authentication has been implemented in elog using the **Kerberos** authentication scheme. This widely used system is also used in MS Windows Domain Controllers, and can be used for site logins, meaning that the same credentials can be used on all computers of a site.

To use that authentication, Kerberos has to be installed on the server running the elogd daemon. Please read the Kerberos documentation how to do this or talk to your site administrator. There are packages for Linux, Windows and Mac OSX. If you compile the elogd program yourself, make sure to have the flag **** HAVE_KRB5 >** defined in the compilation process. To configure elogd to use Kerberos, use following options:

4.12.1 Authentication = <method(s)>

4.12.2 Kerberos Realm = <realm>

where <method(s)> can be **File** or **Kerberos** or both such as in **Kerberos, File**. If the authentication option contains **Kerberos**, the user credentials are authenticated using the default Kerberos Realm. This is typically obtained from the file `c:\windows\krb5.ini` (Windows) or `/etc/krb5.conf` (Linux). If another than the default realm should be used, this can be overwritten with the **Kerberos Realm** option.

When Kerberos authentication is used, the password file is still used to store additional user information such as the full name and the email address, but the authentication is done via the Kerberos server.

If both authentications **Kerberos, File** are enabled, the credentials are first authenticated via the Kerberos server, and - if not successful - via the password file. This allows combined elog installations with centralized and local elog accounts. If the Kerberos authentication was successful, the password in the password file is overwritten with the encrypted Kerberos password. This allows the system to work even if the Kerberos server is temporarily not accessible.

If the password is changed via the "Change Password" button on the config page, the system tries to change the password in the Kerberos database. On some installation it has been found that this does not work, in which case you have to change your password by other means (such as via the Windows login if you use a Windows Domain).

Beside the Kerberos authentication, elogd version 3.0 and higher can be configured to accept a authentication done by the webserver.

4.12.3 Authentication = Webserver

You can also combine it with other authentication methods as shown for Kerberos.

Elogd is then accepting the username set in the Request-Header "X-Forwarded-User" as already logged in. To make this work, you need to configure the webserver correctly, as describe in the adminguide.

4.13 LDAP authentication

LDAP (lightweight Directory Access Protocol) has been implemented by vykozlov in a separate branch at <https://github.com/vykozlov/elog-ldap>. The code has been merged into this distribution on an as-is basis. Following info has copied from the link above:

To use LDAP authentication, do the following:

- Enable LDAP authentication in the **Makefile** by setting **USE_LDAP = 1**
- Change `elogd.cf` to contain LDAP authentication:
 - **Authentication = LDAP**
 - **LDAP server = ldap://example.org:389**
 - **LDAP userbase = ou=People;dc=example,dc=org**
 - **LDAP login attribute = uid**
 - **LDAP register = 1**

The **login attribute** is from the DN (distinguished name), e.g. `uid=user,ou=People,dc=example,dc=org`. The **register** flag determines if LDAP users are automatically stored in the local password file, which is necessary for email notifications.

Please note that it is not possible to change a password in the LDAP database from within ELOG.

4.14 PAM authentication

PAM (Pluggable authentication modules) support has been implemented by Jan Christoph Terasa. To use PAM in `elogd`, do the following:

- Compile `elogd` with PAM support, by either setting `USE_PAM = 1` in the `Makefile`, or by specifying it when invoking `make`. If you compile via CMake, set `USE_PAM` via `ccmake`.
- Enable PAM authentication in `elogd.cfg`:
- `Authentication = PAM`
- `Password file = elogd.passwd`
- `Self register = 3`

The `Password file` is used to store the user names and email addresses of PAM authenticated users, since this information can not be (universally) requested via PAM. For security reasons the password file does **not** store a hash of the user password. Self registration has to be enabled (`Self register ≥ 1`) to use PAM authentication. - To be able to use PAM, the PAM module in `elogd` needs to be able to access the authentication facilities on the system (e.g. be able to read `/etc/shadow`). This can be achieved by either running `elogd` as `root`, or by specifying the appropriate SUID/GUID values for the binary. **[WARNING:]** When running `elogd` as `root`, be careful when using the `-x` option to enable execution of commands via `$shell`, since the commands will be executed using the access rights of the user running `elogd`!

Please note that it is not possible to change the PAM password within ELOG. Instead, please use the available methods on the system

4.15 EMail notification

- `Email <attribute> <value> = <list>`
- `Use Email Subject = <string>`
- `Use Email Subject Edit = <string>`
- `Use Email From = <string>`
- `Default Email From = <string>`
- `Use Email Heading = <string>`
- `Use Email Heading Edit = <string>`
- `Omit Email To = 0|1`
- `Suppress Email to users = 0|1`
- `Email attributes = <list>`
- `Use Email URL = <URL>`

To send email automatically when new entries are created in a logbook, a `SMTP host =` entry must be present in the `[global]` section of the configuration file. To submit an email based on an attribute value, use the statement `Email <attribute> <value> = <list>`. Whenever an entry is submitted where `attribute` is equal to `value`, an email notification is sent to the email addresses in `list`. Several mail addresses may be supplied, separated by commas. The mail addresses can contain attributes via the `"$"` substitution. If a logbook contains for example an attribute `name` which contains email names, then one can put `!$name@domain` to form a valid email address.

Multiple `Email xxx` statements may occur in a configuration file. If either the attribute or the value contains one or more blanks the string must be enclosed with quotation marks, as in:

4.15.1 Email type "Normal routine" = ...

The statement `Email All = <list>` sends an email notification independent of the type and category. The `Use Email Subject = <string>` statement specifies which text is used as the email subject. The text can contain `$(attribute)` statements which are substituted with the current value of that attribute. For a full list of possible substitutions, see the `"Subst <attribute>"` option. The `Use Email Heading = <string>` specifies the text for the email heading line. Default is `"A new entry has been submitted on [host]"`. The option `Use Email Heading Edit = <string>` works the same way for updated (edited) entries.

The option `Use Email From = <string>` is used for the `"From:"` field in the email. Since more and more email servers do not accept invalid `"From:"` addresses in order to reduce spam mail, it might be important that a "real" email address is used in the `"From:"` field. If `Use Email From` is present, it

is always used. If not, the email address of the currently logged in user is used for the "From:" field. If no user is logged in, or the current user has not specified a email address in the password database, the setting of the option **Default Email From** is used for the "From:" field. Only if this option is not specified, a generic address *ELOG@<hostname>* is used, which might be rejected by the SMTP server however.

If the flag **Omit Email To** is set to **1**, the *To:* field in the email is left empty instead set to the real email address of the recipients. This can be useful if one recipient should not see the email addresses of the other recipients.

The flag **Suppress Email to users** can be set to **"1"** if email should only be sent to the recipients of the **Email <attribute> <value> = <list>** statements but not to the users who have registered for automatic email notification.

If one wants to send only some attributes but not all in an email notification, one can use the option **Email attributes = <list>**, where a subset of the attributes can be specified as well as their order. [{#flags}]

The option **Use Email URL = <URL>** can be used to set the URL of the ELOG logbook used in email notifications. This can be useful if no **URL = ...** statement is used for some reason.

4.16 Flags

4.16.1 Show text = 0|1

This flag controls if logbook entries contain a body text. If an installation only requires attributes, this flag can be set to **0**. Default is **1**.

4.16.2 Enable attachments = 0|1

This flag controls the attachment submission at the bottom of a message entry page. If this flag is **0**, the attachment section is not displayed. This might be useful for logbooks where attachments are not used. Default is **1**.

4.16.3 Show attachments = 0|1

This flag controls the display of attachments such as images on normal logbook pages. For logbooks with large images, this flag can be turned off, so that attachments are only displayed when they are clicked on. Default is **1**.

4.16.4 Preview attachments = 0|1

This flag controls the display of attachments in the edit form. If one uploads an attachment, but has not yet submitted the entry, the uploaded attachments are shown at the bottom if this flag is **1**. Only ASCII files and images are shown of course. Default is **1**.

4.16.5 Summary lines = x

This specifies the number of text lines displayed in a summary page. Zero displays no text at all. The default is **3**.

4.16.6 Summary line length = x

This specifies the number of characters of the summary lines. After this number of characters, a line break is inserted in long lines to keep the column width not too wide. The default is **40**.

4.16.7 Attachment lines = x

This specifies the number of text lines displayed for ASCII attachments. For long ASCII attachments, it can be useful to only display the first few lines not to make the HTML page too long. The default is **300**.

4.16.8 Reverse sort = 0|1

If this flag is **1**, all listing pages (the default page view, the result of a search query and the result of the "Last day" query) is sorted in reverse order (newest entry down to oldest). The checkbox *Sort in reverse order* on the search form gets checked by default, too. Sorting in reverse order can make sense if there are many pages of entries, but the ones entered last should be displayed on the first page. Default is **0**.

4.16.9 Search all logbooks = 0|1|2

If this flag is **1** or **2**, the search form displays the button "*Search all logbooks*". If the flag is **2**, the button is checked by default. Setting this flag to **0** hides this button. It might be necessary to do this for public logbooks if there are also protected logbooks. Otherwise the search result would also display entries from the protected logbooks. The default is **1**.

4.16.10 Enable browsing = 0|1

If this flag is **1**, browsing (hitting the next/previous button) is enabled. For some rare occasions it might be necessary to disable browsing. Default is **1**.

4.16.11 Filtered browsing = 0|1

If this flag is **1**, browsing (hitting the next/previous button) can be filtered by individual attributes. If the checkbox next to an attribute is checked, only messages with the same attribute value are displayed. Default is **0**.

4.16.12 Default encoding = 0|1|2

This specifies the default encoding for new entries. For installations where entries are normally submitted as plain text, the default can be set to **1**. Set to **0** for [ELCode](#) encoding, to **2** for HTML encoding. The default is **2**, which activates the built in FCKeditor automatically for new installations. If this editor is not wanted or people are concerned about cross site scripting, the default encoding should be set to **0** or **1**.

4.16.13 Allowed encoding = <n>

Allowed encoding options. **<n>** can be the sum of following flags: - **1** : Plain - **2** : [ELCode](#) encoding - **4** : HTML encoding

To allow plain and HTML encoding for example, set **<n>** to 5. Default is **7**. Note that allowing HTML encoding may cause some security risk, since an elog entry may contain malicious scripting code. It should therefor only be allowed for installations where it is really needed and with no public write access.

4.16.14 Allow HTML = 0|1

This flag allows or denys the usage of HTML in attributes. Note that allowing HTML encoding may cause some security risk, since an elog entry may contain malicious scripting code. It should therefor only be allowed for installations where it is really needed and with no public write access. The default value is **0**.

4.16.15 Suppress default = 0|1|2|3

This specifies the default state of the "*Suppress Email notification*" button on the new message entry form. For installations where normally an email notification is not necessary, the default can be set to **1**. If an important entry is entered, users can then uncheck the suppress box. If this value is set to **2**, the suppress box is not displayed at all, so that an email notification is always produced. If this value is set to **3**, the email notification is always suppressed. The default is **0**.

4.16.16 Suppress Email on edit = 0|1|2|3

This is the same as **Suppress default**, but just for edited entries. The default is **0**.

4.16.17 Resubmit default = 0|1|2

This specifies the default state of the "*Resubmit as new entry*" button on the edit message entry from. If this button is checked, the current message is removed from its current position in the database and submitted as a new message. This can for example be useful for applications where users want to see which records have been updated recently. If this value is set to **2**, the resubmit box is not displayed at all. The default is **0**.

4.16.18 Resubmit replies = 0|1

If this flag is set to **1** and an entry is resubmitted as a new entry and this entry has replies, all replies of this entry are resubmitted as new entries as well. The default is **0**.

4.16.19 Display Email recipients = 0|1

If this flag is **1**, the email recipients are displayed when a logbook entry is entered which produces an email notification. Setting this flag to **0** suppresses this display, in case users need not see that email is being sent and to whom. The default is **1**.

4.16.20 Email Format = <n>

Specifies what is sent in an email notification. <n> is the sum of following flags:\ - **1** : Send heading line "A new entry has been submitted..." - **2** : Send attributes - **4** : Send URL of logbook entry - **8** : Send message body - **16**: Send optional attachments as email attachments - **32**: Send logbook name - **64**: Send names of optional attachments

So to send for example only the attributes and the URL, set <n> to **6**. Default is **63** (send everything).

4.16.21 Email Encoding = <n>

Specifies in which encoding an email is sent. <n> is the sum of following flags:\ - **1** : Plain text - **2** : HTML in the form of the plain text, but with ELCode interpreted - **4** : Full HTML page as shown in elog

So to send email in plain text and full HTML, set <n> to **5**. Some email clients have the possibility then to switch from one view to the other. Default is **2**.

4.16.22 Max email attachment size = <n>

This option specifies the maximum allowed email attachment size for email notifications. Most mail delivery systems have a maximum attachment size and refuse to accept emails with larger sizes. If the size of an attachment exceeds this limit, it is not included in the email notification but rather a link to the attachment on the elog server is used. The default value is **10000000** (ten million bytes).

4.16.23 Back to main = 0|1

If this flag is **1**, the "*Lis*" button takes you back to the logbook selection page instead to the last entry of the current logbook. The default is **0**.

4.16.24 Logout to main = 0|1

If this flag is **1**, the "*Logout*" operation takes you back to the logbook selection page instead to the login page. The default is **0**.

4.16.25 Logout to URL = <URL>

If this URL is set, the "*Logout*" operation takes you to a specific web page specified in the URL.

4.16.26 List after submit = 0|1

If this flag is **1**, the list page is shown after the submission of a new entry. If this flag is **0**, the entry just submitted is shown. The default is **0**.

4.16.27 Restrict edit = 0|1

If this flag is **1**, users can only edit their own messages. The system checks automatically if the currently logged in user matches the user supplied in an author attribute via the "*Preset xxxx*" option. The default is **0**.

4.16.28 Expand default = 0|1|2|3

This setting determines how messages are displayed in threaded mode. Following options are possible: - **0**: Only message heads are displayed, no replies. A "+" indicates which message has one or more replies. - **1**: Messages and replies are displayed, but no message body. - **2**: Messages and replies are displayed together with the first few lines of the message body. The number of lines is controlled by the **Summary Lines** option. - **3**: Messages and replies are displayed together with the full message body.

The default is **1**.

4.16.29 Hidden = 0|1

If this flag is **1**, the logbook is not displayed in the initial logbook selection page and in the logbook tabs. This can be useful for logbooks which are only accessed for backup or archiving and would clutter up the logbook list for the normal user. To access hidden logbooks, one has to enter the logbook URL directly, or from a bookmark list. Default is **0**.

4.16.30 Hide Comments = 0|1

If this flag is **1**, the logbook "Comment" is not displayed in the logbook selection page. Default is **0**.

4.16.31 Use Lock = 0|1

If this flag is **1**, a logbook entry is *locked* when someone edits it (clicking the *Edit* command). A locked message gets displayed with a little red sign indicating that the message is currently edited by someone and should not be touched. This can be helpful in installations where several people can edit messages. Without locking, the second submission of an edited message overwrites the first submission without notice. Although the sign gets displayed, the message can still be edited (the lock can be "stolen"), but it's the user's response to avoid any conflict.

Since elog cannot determine if someone keeps a message very long for editing or if only the browser got closed, the locking can show up even if the message is not kept for editing any more. In that case, the message has to be edited again and submitted, to remove the original lock.

Note that logbooks accessible from the internet usually get scanned by search engines. This can lead to situations where the *Edit* link of each message is "followed" by a bot, resulting in all messages being locked. In those cases locking has to be turned off.

Since release 2.5.4, some Javascript code has been added to avoid unwanted locks. If someone edits an entry, but then goes away from that page or closes the browser without submitting the changes, a pop-up window appears asking the user to submit the changed entry. Although this works for most browsers in most cases, it could be that Javascript has been turned off in a browser, in which case the stale locks still might appear.

Default for "Use Lock" is **0**.

4.16.32 Show top groups = 0|1

When using top groups, the root of the elogd server is not accessible any more, to avoid cases where one group can "see" the logbooks of the other groups. If this feature is unwanted, the flag **Show top groups** can be set to **1**, in which case a list of available top groups is shown.

4.16.33 Fix text = 0|1

With this options the main text body can be fixed, so that it cannot be changed via the **Edit** button later. This feature can be useful for set-ups where some attributed must be changed later, but the text body should be preserved. The default is **0**.

4.16.34 Case sensitive search = 0|1

This switch has two meanings. First, it defines the default state of the **Case sensitive** check box in the "Find" page. Second, it determines if the quick filters are case sensitive or not. The default is **0**.

4.16.35 Mode commands = 0|1

If this flag is missing or set to **1**, the links "Full", "Summary" and "Threaded" are shown on the top of the listing page. If this flag is set to **0**, these commands are hidden. This might be useful in logbooks where only one mode makes sense for example.

4.16.36 Suppress execute default = 0|1

External scripts can be called with the **Execute new/edit/delete** options. If these options are enabled, a checkbox appears which lets the user suppress execution of the external script. The setting of this flag determines the default state of this checkbox. In logbooks where a script should only be occasionally executed, it could make sense to set this flag to **1**.

4.16.37 Preserve IDs = 0|1

When a logbook entry is copied or moved to another logbook, it obtains a new entry ID in the destination logbook. This can cause problems if the logbook entries reference each other with their IDs. To keep the same ID in the destination logbook, this setting can be set to **1**. If an entry with the same ID in the destination logbook exists already, it gets overwritten. Default for this setting is **0**.

4.16.38 Collapse to last = 0|1

In threaded view, the list of replies can be collapsed into a single entry. If this flag is **1**, then the last entry of each thread is shown, otherwise the first thread is displayed. Default for this setting is **1**.

4.16.39 Sort Attribute Options <attribute> = 0|1

If this option is **1**, the options for this attribute are sorted alphanumerically. This can be handy when locating options from long lists in drop-down boxes in quick filters for example. Default for this setting is **0**.

4.16.40 Allow branching = 0|1

With this option one can prohibit "branching", which is that an entry gets more than one reply. When branching is prohibited, only linear threads are possible, which is one head entry, one reply to it, then one reply to the reply and so on. Default for this setting is **1**.

4.16.41 Enable Smileys = 0|1

When encoding an entry with ELCode, certain sequences such as :-) get automatically converted into small "smiley" images. If this behavior is not wanted, it can be turned off with this option. The default for this setting is **1**.

4.16.42 Refresh = <seconds>

The elog listing page can be refreshed periodically with this option. If it is given, the page automatically reloads after <seconds>. This can be useful for logbooks where other people often post entries or where some entries are posted automatically (via the elog utility) and one wants to keep an eye on what's new. The default for this setting is **0** meaning no refresh.

4.16.43 Show last default = <days>

In large logbooks, search operations can take quite long, blocking other users from accessing ELOG. On the *Find* page, one can restrict the search operation to a certain time period, like last day, last week, etc, which makes searching much faster, but restricts it to a certain time in the past. If one forgets however to enter anything in the *Show last* drop-down box, then the search again can take quite long. This option pre-selects an option in the *Show last* drop-down box, so that the user does not have to think about selecting a certain time period. Following options are possible: **0, 1, 3, 7, 31, 92, 182, 364**. "0" means an unrestricted search default.

4.16.44 Save drafts = 0|1

Starting with version 3.1, ELOG supports auto saving. When text for a new entry is entered in the browser, it might get lost if the browser windows is closed before the entry has been submitted. In order to avoid this, entries can be saved as drafts, to be finished and submitted later. This can be achieved by clicking on the **Save** button or by the *autosave* feature (see next option). The **Save drafts** option turns this feature on or off. Default is **1**.

4.16.45 Autosave = <seconds>

Drafts can be sent to the server regularly after some editing (see previous option). This option determines the interval this is done. The default is **10** seconds after the last edit. Setting Autosave to zero disables the autosave functionality.

4.16.46 List drafts = 0|1

By default, draft entries are shown in the list display in another browser when the entries are currently edited. This can be confusing to other users since the draft entries are frequently updated. To avoid this, this flag can be set to `0`, which hides all draft entries in the list view. If they are hidden, the only way to come back to them is to hit the **New** menu item, in which case the system presents to the user a list of open draft messages to be continued.

4.16.47 Hard wrap = 0|1

If entries are entered in plain text mode, the browser adds automatically a CRLF at the end of each line where the text wraps. This ensures that the submitted entry has the same line breaks as in the edit box. If this behaviour is not wanted, the adding of hard wraps can be turned off by setting this value to `0`. If the user then enters a very long line without hitting the newline key, the long line is preserved which can make it hard to read.

4.17 Themes

Themes are layout and color schemes which determine the look and feel of a logbook (sometimes called "*skins*"). A theme consists of a set of images, which are used for the title banner and browse buttons, and a Cascading Style Sheet (CSS), which defines the colors, fonts and spacing of the ELOG pages.

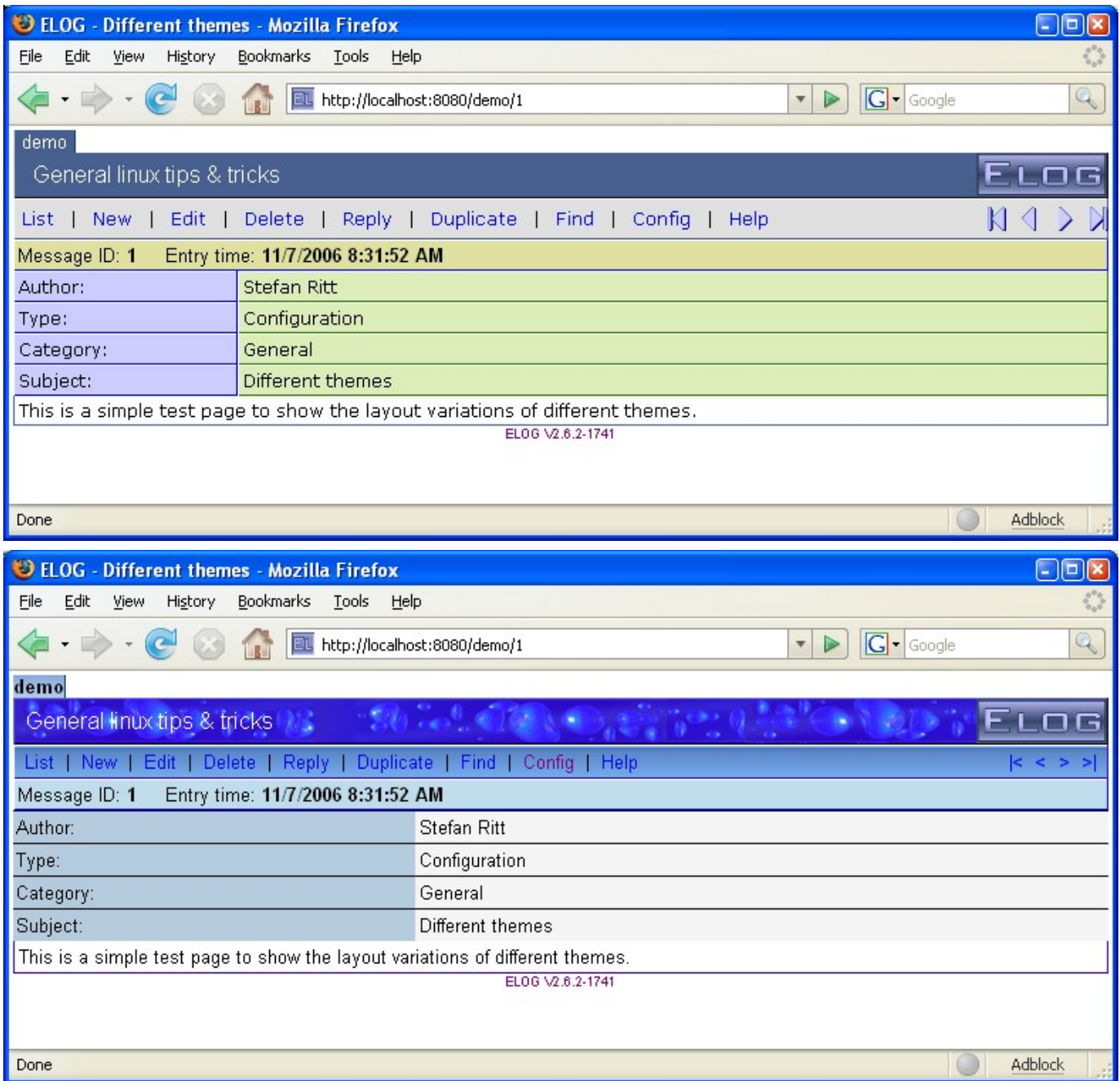
Each theme resides in a separate subdirectory and is specified with the `theme = <dir>` option in the configuration file. Each theme can contain several CSSs, which can be selected with the `css = <filename>` option.

A default theme is contained in the distribution. If new themes are developed by users, they can be sent back to the author, to be included in future releases.

To change colors and fonts, the source of a ELOG page can be examined. All elements use CSS classes which are specified in the `class="<name>"` statements. These classes can be found in the `.../themes/default/default.css` file and changed accordingly. For a description of all options, please consult for example the [W3C](#) consortium.

If the CSS file is edited, most browsers require a "reload" to refresh the modified file. The **elogd** daemon does not have to be restarted after a change in the CSS file.

These two images display the same logbook entry using different themes:



4.18 Mirroring

Sometimes it can be useful to have the same ELOG logbook on two different computers. This might be the case if you travel with your laptop, but want to keep the logbooks from your desktop computer on the laptop. The problem is that if you add an entry on your laptop, the logbooks on the laptop and the desktop get out of sync. Merging only the ELOG database files does not help, since two entries could be made at the same day on the laptop and the desktop, which would lead to a conflict in that day's database file.

To solve this problem, *mirroring* was introduced from Version 2.5.0 on. This technology allows to synchronize one ELOG server with a number of other servers on a per-entry basis. No additional software is needed, only two `elogd` daemons talking to each other. The synchronization can be executed manually or periodically. If entries are changed/added/deleted on both sides, they get merged properly during synchronization. In order to minimize network traffic, each ELOG server calculates a MD5 checksum for each message, which gets exchanged during synchronization. Only when the MD5 checksum differs, entries are transferred.

To set-up mirroring, install two elogd servers on two machines (for testing purpose that also works on one machine with two elogd servers running on different ports). This can be done in two ways:

1. Automatic configuration

A complete elog server can be transferred to a secondary server using the `clone` command. Assume the existing server resides at `http://master.your.domain/`, and you want to mirror this server to a new location at `http://slave.your.domain/`. You do that by installing the `elog` package at the slave machine, and then executing on the slave:

```
`elogd -C http://master.your.domain`
```

or

```
`elogd -C https://master.your.domain`
```

for a remote server running under the SSL protocol. Note that you have to put "Allow clone = 1" temporarily into the `elogd.cfg` file of your existing server to allow cloning. This opens a password-free access to your existing server, so remove it immediately after you finished cloning.

This command tells `elogd` to retrieve the configuration file, and optionally all logbook entries and password files from the master machine. Note that both servers must be version 2.5.4 or later. In case of trouble, you can turn on verbose messaging:

```
`elogd -v -C http://master.your.domain`
```

which could give some hints. If a logbook on the master server uses restricted access, you have to specify the admin user name and password. After everything has been transferred, you can start `elogd` in the normal way.

2. Manual configuration

First, copy the `elogd.cfg` file from the master to the slave server. Make sure that the files are identical (except the port setting if you run two servers on the same machine). Then, add the following configuration options. They should be put into the `[global]` section of the configuration file:

4.18.1 Mirror server = <URL-list>

This statement specifies one or more mirror servers. Each URL must contain the host, port and possible subdirectory of the remote server, as if you would access it through your browser. A typical statement looks like:

```
Mirror server = myhost.mydomain.org:8080, http://another.server.org/elog/, https://yet.another.org
```

The URL should not contain any logbook name, this gets added automatically. The second example contains a subdirectory, which is typically used if the `elogd` daemon runs under an Apache proxy. The third example shows a server running under the SSL protocol.

4.18.2 Mirror config = 0 | 1

Normally, only the logbook entries are mirrored. One can also mirror the contents of the `elogd.cfg` configuration file for individual logbooks. This can be turned on by setting this option to `1`. Default is `0`. Only the individual logbook section is mirrored, not the `[global]` section. Settings which are specific to one server, for example the `URL =` statement, should then be kept in the `[global]` section, so that they are not mirrored between different servers.

4.18.3 Mirror cron = Minute Hour Day Month Weekday

This statement turns on periodic mirroring. The format is similar to the UNIX `cron` command. Each of the five values can either be an asterisk, which means all possible values, a comma-separated list or a range. It can be explained most easily with examples:

Mirror cron	meaning
0 3 * * *	Every night at 3:00
30 7 1,15 * *	At 7:30 every 1st and 15th of a month
0 12 10 10 *	Once a year at 12:00 on my birthday
0 7-18 * * 1-5	Once every hour from 7:00 to 18:00 from Monday to Friday

Valid ranges for each value are:

Item.	Range
Minute	0-59
Hour	0-23
Day	1-31
Month	1-12
Weekday	0-6 with 0=Sunday, 1=Monday, etc.

If mirroring is turned on, it is advisable to use the **Logfile =** option to turn on logging, so that one can inspect the logfile to see if the mirroring works correctly.

4.18.4 Mirror user = <name>

If periodic mirroring is used via the **Mirror cron =** statement and the remote logbook uses user-level access, this statement specifies the user name which is used to log in to the remote logbook. The password is taken from the local password file and has to match the password in the remote password file, otherwise the access is not allowed. The user name is typical the login name of the administrator.

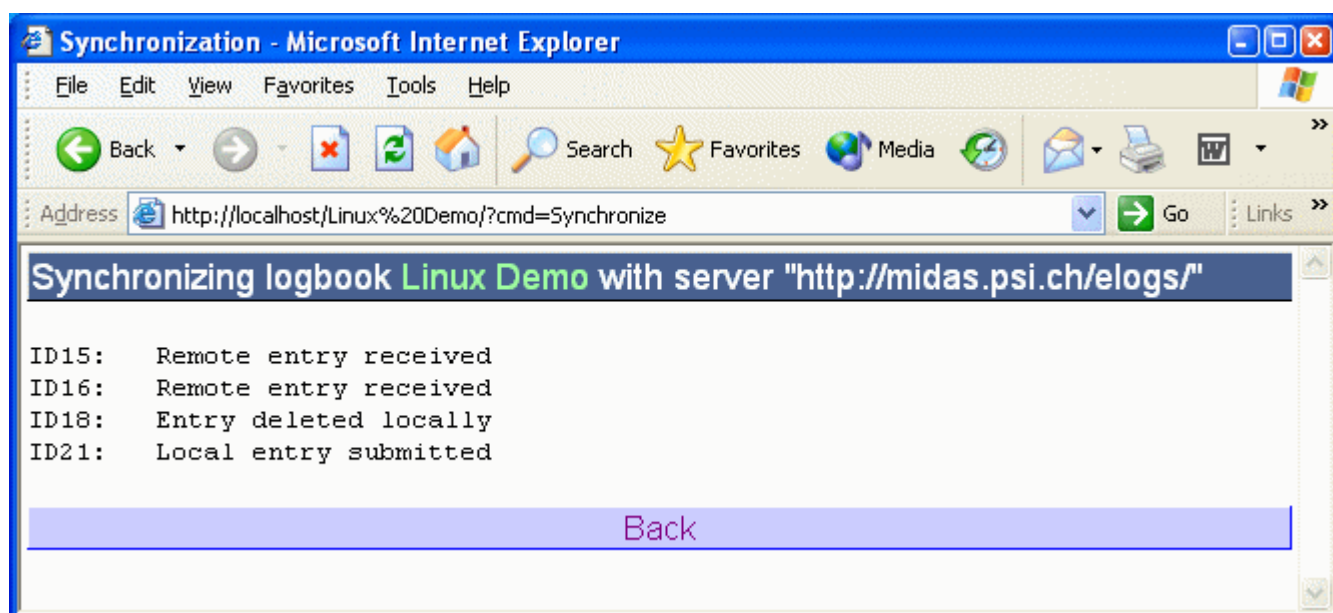
4.18.5 Mirror simulate = 0 | 1

If one wants to try out mirroring without causing any harm, one can turn on this flag. During synchronization, entries are compared and necessary transfers are displayed, but not executed. Default is **0**.

4.18.6 Mirror exclude = 0 | 1

By default, all logbooks are mirrored. Individual logbooks might be excluded from mirroring by putting **Mirror exclude = 1** in their individual logbook section of the configuration file (**Not** the [global] section). Default is **0**.

If the statement **Mirror server** is present in the configuration file, a new menu option "**Synchronize**" appears on the elog page. Clicking on this menu options starts the synchronization:



On the left side one sees the entry ID's. Entries which are equal locally and remotely are not displayed. Here are the rules for synchronization:

- If an entry has been modified locally but not remotely, it is submitted to the remote server.
- If an entry has been modified remotely but not locally, it is retrieved from the remote server and saved locally.
- If an entry has been modified remotely and locally since the last synchronization, an error is shown that the entries are conflicting. In that case one has to merge the entries manually and delete it on one side.
- If an entry has been deleted locally, it is deleted remotely.
- If an entry has been deleted remotely, it is deleted locally.
- If a new entry exists locally, it is submitted.
- If a new entry exists remotely, it is retrieved from the remote server and saved locally.
- If new entries exist locally and remotely having the same entry ID, the local entries are changed to have higher entry IDs, then the remote ones are retrieved. Care should be taken if external links (such as `eLog:123`) to the local entries are used, since they will point afterwards to the wrong entry.

By starting the synchronization on one elogd server, this server becomes the client and the other one becomes the server. This means that the local server actively compares the local and the remote messages, and updates one or the other if necessary. The other (remote) server does not need to have any mirror option in its configuration file, since the local server simulates a web browser to send and retrieve messages to the remote server. It is however allowed that the remote server also contains some mirror settings in the configuration file, this way the synchronization can be started from both servers.

5. ELOG FAQ

Frequently Asked Questions about usage and configuration

Please check also the [ELOG Forum](#).

5.1 How does one configure elog to display the last message by default

One can use the `Start page` option for that. The entry:

```
Start page = 0?cmd=last
```

shows the last message entry by default. To have the described behaviour for all logbooks, the above statements can be placed in the `[global]` section.

5.2 Are there any plans to implement a MySQL back end?

No. The idea behind **ELOG** is that it is a *simple to use, simple to install* application. Many people use **ELOG** under Windows, and they even don't know what MySQL means. Other people like the flat file database format, because it's simple, easily accessible from other programs, and it's easy to backup certain days or months of the database (since the filenames contain the date). Since **ELOG** should be independent of any other package, some "switchable" backend between native **ELOG** format and MySQL would be needed, which is lots of work and not planned right now.

However, there are several contributions from other people who wrote scripts to put ELOG entries into a MySQL database. One is available at <http://elog.psi.ch/elog/Forum/387>.

5.3 Can I run the ELOG daemon from inside Apache or any other Web server?

No. The **ELOG** daemon was designed as a standalone server and it will stay like that in the future. The reason for that is that `elogd` should not rely on any other software. This is for example important for many people running `elogd` under Windows, and they have no clue how to install Apache for Windows. The installation and maintenance for `elogd` therefore becomes much simpler. To run `elogd` in parallel to an Apache server on port 80, use Apache as a proxy, following the instruction on the installation page ("Running elogd under Apache").

5.4 I can access my logbook without any password, isn't that a security problem?

By default, no password is used in **ELOG**. This can be useful for public directories etc. that anybody should be able to read. To add password security, read the documentation under [Access control](#). The recommended setup is password file security with guest access.

Note that passwords are transferred over the network in plain text and therefore not secure. If this is a problem, a [secure](#) network connection should be used.

5.5 I want a bookmark pointing to the last page where an attribute has a certain value

Use the URL:

```
http://<your.host>/<logbook>/?cmd=Last&<attribute>=<value>
```

This executes the *"Last"* command using a filter with `<attribute>=<value>`. The following command displays the same page, but also locks the attribute (checks the box next to `<attribute>`) so that browsing (next, previous, first, last) only shows pages with that attribute value.

```
http://<your.host>/<logbook>/?cmd=Last&<attribute>=<value>&l<attribute>=1
```

Note the `"l"` before the second attribute, as in `"*lAuthor=1*"`.

5.6 I want a logbook with public read access (no password), but restricted write access

In an old version of the FAQ it has been stated here that one has to use two logbooks pointing to the same data directory. From Version 2.0.6 on, this can be accomplished much easier by the usage of the **"Guest menu command"**. Use a logbook with user level access (password file), and add menu lists like in the following example to the configuration file:

```
Menu commands = New, Edit, Reply, Find, Last 10, Change password, Logout, Help
Guest menu commands = Find, Last 10, Login, Help
```

If users access the logbook without supplying a user name, they are treated like "guests" and see the "Guest menu commands", with which one cannot submit or edit logbook entries. If one hits the "login" button, a user can login with a user name/password and sees the normal menu commands, with which one can submit new logbook entries.

An optional self registration is possible by specifying

```
Self register = 1
```

in the configuration file. New users can then create their own accounts.

5.7 I have many logbooks with password files, so if I add a user or want to change a password I have to do this for all logbooks which is painful.

You can have several logbooks point to the same password file. So if you change a user or password in that file, it becomes automatically available in all logbooks which use that file.

5.8 How can I configure ELOG such that it displays something else than the message list by default?

There is a simple trick. You use the **"Start page"** option in the **eLogd** file to redirect the start page to something else. Here are some examples:

```
?npp=5           for the last 5 messages
?last=7          show last 7 days (week)
?cmd=New         show the new message entry form
?cmd=Find        Show the "find" page
?cmd=Search&<attrib>=<value> for a search with <attrib>=<value>
```

The various URLs can be copied from the browser's address bar when doing various things there.

5.9 I want to have additional commands specific to my logbook

New commands can be added for example with the **"Bottom text = bottom.html"** option. To display all messages from last week and month of with "category = info", one can put following HTML code in bottom.html:

```
<center>
<a href="?last=7&Category=Info">Info from last week</a> |
<a href="?last=31&Category=Info">Info from last month</a> |
</center>
```

Note that the parameters **"last=7&Category=Info"** applies a filter on the display. You can learn how to make these filters by looking at the URL in your browser when you submit a find command with certain options.

5.10 How does one configure elog to disable editing of existing messages? I want a logbook where one can enter messages but not change them afterwards.

This works with the option **Menu commands**. By default, the menu commands **Back, New, Edit, Delete, Reply, Find, Config, Help** are displayed and allowed. To avoid editing (and deleting) of existing messages, one removes the two commands and puts following statement into **eLogd.cfg**:

```
Menu commands = Back, New, Reply, Find, Config, Help
```

This prohibits the execution of the commands "Edit" and "Delete".

5.11 How can I track various revisions of a message using the "edit" command?

There are two ways:

- Add an attribute which keeps the revision dates and names as follows:

```
Attributes = Author, ..., Revisions
Locked attributes = Revisions
Subst on Edit Revisions = $Revisions<br>$date by $long_name
```

The "Revisions" attribute cannot be modified manually (since it's locked). On each edit, the date and the current author is appended to the previous revisions. The "\
" puts a line break between the entries.

- Create a thread for each entry. In addition of having the date and author of different revisions, the message content is kept. To revise an entry, one hits "Reply" instead of "Edit" (one can disable the "Edit" command for example). If one puts following option into the configuration file:

```
Reply string = ""
```

then the reply contains the original message without the usual "> " at the beginning of each line. One can then edit the message and submit it. In the threaded message list display, one sees then the different revisions as a message thread.

5.12 How can I enter a date which is different from current one?

Usually, the current date/time is recorded when you add a new entry. It might be, however, that one wants to enter "old" entries, or some entries with a date in the future (like a to-do list with a due date). To do that, one can add a new attribute (let's call it *Record date*, to be different from the pre-defined *Date*):

```
Attributes = Author, ..., Record date
Type Record date = date
Preset Record date = $date
Date format = %Y %m %d
List Display = Record date, Author, ...
Start page = ?rsort=Record date
```

The *Preset Record date* statement sets the record date to the current date, but this can then of course be edited during the message entry. The *List Display* and *Start page* statements show the record date as the first column in the summary display and also sort by that. Note the *Date format* showing first year, then month and day. This is necessary since sorting is done only lexically. Please note that the *List Display* was renamed recently. Prior to version 2.3.10, it was called *Display Search*.

5.13 I cannot pass the login page, it's always redisplayed even if I put in the right password?

This can happen if you change the login policy, for example move the `Password file =` entry in the configuration file from a logbook section to the [global] section or back. In that case some old cookies could be stored in your browser, which confuse the system. Please delete your cookies in the browser to resolve this problem. Read your browser documentation on how to do that.

5.14 How can I change an attribute for an entire thread? We have an attribute "open problem/fixed" which should be changed for the whole thread if that problem has been fixed.

This is a typical request of a bug-tracking set-up. Someone enters a request, opening a new thread. The expert(s) reply to the request, and after a while, the request gets satisfied or the problem gets fixed. If an attribute like "status", having the options "open" and "fixed" could get changed for the whole thread, one could very easily search for all "open" problems.

Since this functionality is not implemented, an alternative strategy is recommended: Implement two (or more) logbooks. The first logbook has open issues, the second one has fixed ones. When an entry changes state, it simply has to be copied to the second logbook. This can be done by defining the menu command "move" in the config file, like:

```
Menu commands = Back, New, Edit, Delete, Reply, Find, Move to, Config, Help
```

Note the additional **Move to**. This solution is even more elegant than having attributes changed in whole threads, since one has two separate logbooks, and can treat the second one more like an archive, make separate back-ups, or deleting some entries after some time, while keeping the open issues untouched.

5.15 Can I use RSS feeds with password protected logbooks?

RSS feeds normally only work for logbooks which have at least public read access (via the guest menu commands). There is however a way to allow only restricted read access and still use RSS feeds. This is done by adding an additional read password via the `eLogd -r <pwd> -l <Logbook>` command. This password (username may be any) can then be used in an RSS reader for restricted access. One reader which has been successfully used with this kind of authentication is [RSSReader](#).

5.16 How can I make a whole thread open or closed?

Sometime people want to mark a whole thread in a way. An example is a to-do list, where they want a special icon on high priority things, and have this icon disappear once the task is finished. This can be easily done with icons. The configuration could look like this:

```
Attributes = Author, Status, Subject
IOptions Status = icon1.gif, icon2.gif, icon4.gif
Preset Status = icon4.gif
Preset on reply status = icon2.gif
Icon comment icon1.gif = Closed entry
Icon comment icon4.gif = Open entry
Thread display = $Author $Subject
Thread icon = Status
```

New entries get an exclamation mark icon for example ("Preset status = ..."). Replies to this entry get a reply icon. Once the thread should be closed, one simply edits the top entry in that thread and changes the icon. The icon1.gif from the distribution is maybe not ideally suited for that, but one could make a green check mark icon for example for that. The "Thread display" and "Thread icon" make this icon appear at the left side of the threaded display.

An alternative approach would be to use two logbooks. The first one receive all new entries ("open items"). Once an entry (with its replies) gets closed, it must be moved manually to the second logbook ("closed items"). This can be done with the "Move To" command (see "menu commands" in config file). This way one nicely separates open and closed items in two separate logbooks. One can still search both logbooks at the same time if one checks "Search all logbooks" in the find page.

Starting from elog version 2.7.7, there now even a third way to do this. With an additional line in the configuration file: `Collapse tn last = 1` (in fact the default, but to be explicit), then when the thread is ready to be marked as closed, select the "closed entry" icon when writing the last entry. When the entries are viewed in "threaded" mode, then the closed entry icon appears on the last entry; and when in "threaded, collapsed" mode, then the closed entry icon appears in the one line that represents that whole thread.

5.17 Does elog have a spell checker?

No, but you can use any spell checker which works with your browser. Examples are [IESpell](#) for Internet Explorer and [SpellBound](#) for Mozilla-based browsers.

5.18 Why are entries with large attachments submitted so slowly?

If email notifications are used, the ELOG program has to pass these attachments to the email server, which might take quite some time. Some email servers even don't allow to forward attachments if they are larger than a few mega bytes. In that one can simply turn off the forwarding of email attachments with

```
Email format = 111
```

this causes only the attachment names being forwarded, not the attachments themselves.

5.19 The elgod daemon crashes from time to time, what can I do?

Bugs are constantly fixed inside elogd so a upgrade to the current version is recommended as a first measrue. If that does not help, the key will be the reproducibility of the crash. I only can fix problems if I can reproduce them. Sometimes it's related to strange logbook entries which cause elogd to crash when they are edited. So if there is a way to reproducible trigger the problem, I need the files and confiration related with it. If I can reproduce it in my local installation, I can fix it pretty soon.

If that is not possible, an alternative is to run elgod under a debugger, and do a stack trace if the program dies. Under linux, this can be done using the gdb debugger, which might look like this:

```
[~/elog]$ gdb ./elogd
GNU gdb Red Hat Linux (6.5-25.el5rh)
...

(gdb) run
Starting program: /afs/psi.ch/user/r/ritt/elog/elogd
elogd 2.7.5 built Dec  2 2008, 10:47:09 revision 2147
ImageMagick detected
Indexing logbooks ... test

Program received signal SIGSEGV, Segmentation fault.
0x08054beb in el_index_logbooks () at src/elogd.c:3892
3892      *p = (char)1;
(gdb) where
#0  0x08054beb in el_index_logbooks () at src/elogd.c:3892
#1  0x080b8774 in server_loop () at src/elogd.c:27565
#2  0x080bbdd5 in main (argc=1, argv=0xbfee5b54) at src/elogd.c:28923
(gdb)
```

So the basic command is to make a stack trace with "where" after a segmentation fault. This tells me where in the code something wrong happened (in this case it was inside the function el_index_logbooks() at line 3892. Please send me this information and I will try then to figure out what was wrong.

5.20 How can I create an ELOG entry automatically from a script?

The [User's Guide](#) describes the standalone "elog" utility, which can be used from a script or from another program to submit an automatic email entry. This works locally or remotely, with optional attachments. Enter "elog -h" for a full list of options. The elog utility is part of the distribution and resides in the same directory as the elogd daemon.

5.21 I want to notify different people for different things, how do I set up this?

Assume you want to send an email notification to person A for a problem report, to person B for problem fix and so on. The simplest way is to use the `Email <attribute> <value> = <email address>` syntax. So you could set-up following configuration:

```
Attributes = Author, Type
Options Type = Problem Report, Problem Fix
Email Type Problem Report = person.a@elog.com
Email Type Problem Fix = person.b@elog.com
```

If you want to select email addresses directly from a list, you can do the set-up as following: Attributes = Author, Notify MOptions Notify = Person A, Person B Email Notify Person A = person.a@elog.com Email Notify Person B = person.b@elog.com This way you can for each entry select one or more people to be notified from the pre-defined list.

6. Forum

A dedicated [discussion forum](#) running elog itself is used for various feedback and questions around elog.

7. ELOG Wishlist

Here are some "*wishlist*" items requested by users. I will work down the list as time permits, starting at the items with the most votes. You can vote for a feature, or suggest one, by sending me an email.

Feature	Votes
Implement groups of users	8
Password expiration	2
Account expiration	1
Make attributes which would be displayed selectable in "Find" page	1
Convert elog text files into XML files and comma separated files with elconv	1
Implement multi-line attributes	8
Derive attribute options from list of entries of another logbook	1
Specify date format explicitly for substitutions, like \$entry date{%Y}	2
Change attributes of whole thread, like mark a thread "open" or "closed"	4
Automatic unlocking of entries after a specific time period	1
Implement attributes which are math results from others, like $att3=att1+att2$	3
Show sums of numeric attributes	2
Let each user choose its language separately	1
Implement multiple selection boxes	1
Select individual columns for display and printing	2
"Incremental" options. Have fixed set of global options, and in each logbook add or remove some of those with +Options = ... or -Options = ...	1
"Include" statement for config file.	2
Use different styles (colors) depending on an attribute	1
Add native IPv6 support	1
Copy new elog entries automatically to other logbooks	1
Extend the search facility to (text) attachments	1

8. ELOG Contributions

ELOG contributions can now be found in a [separate logbook](#)

9. Demo

A dedicated [Demo Logbook](#) is available as a playground.

10. ELOG download information

ELOG is distributed both as source code and as precompiled binaries for various platforms. In addition to major versions, minor releases are made containing bug fixes or some new and not yet completely debugged code. This is usually the case if some user asks for some new features, which are then implemented and sent to the user for testing. The minor releases are named `x.y.z-r` where `r` is the release number. A [web access](#) to the source code contains the complete development history of ELOG, plus the newest fixes and features which might yet be in a release. To check out the Git repository, use:

```
git clone https://bitbucket.org/ritt/elog --recursive
```

No tags are used, so it is recommended to always use the newest release from the "master" branch.

Building `elogd` requires the CMake system and is done in the traditional way:

```
$ cd elog
$ mkdir build
$ cd build
$ cmake ..
$ make
```

This will put the executables `elogd` and `elog` into the build directory, from where they can be moved to a system directory like `/usr/local/sbin/elogd`

News for each version can be seen in the [changelog](#)

10.1 Binary packages

Various binary packages are kept in a separate [download repository](#).

10.2 Windows Binaries

The windows binaries are distributed with an automatic [installer](#). Execute the installer to install ELOG and to register the `elogd` server as a windows service. Previous windows versions can be found [here](#).

Note that the windows binaries are very much outdated and will be updated once the develop gets access again to a Windows PC.