

# The LHCb Electronic Logbook

## Public Note

Issue: Final  
Revision: 1

Reference: CERN-LHCb-2005-073  
Created: August 26, 2005  
Last modified: September 28, 2005

Prepared by: Tobias Lohe and Eric van Herwijnen





## Abstract

This report describes the Electronic Logbook which will be used by the LHCb experiment. The logbook is based on the Open Source software ELOG [2]. The report covers the installation, configuration and adaption of the logbook software as well as the PVSS framework component fwLog [4].

## Document Status Sheet

<b>1. Document Title: The LHCb Electronic Logbook</b>			
<b>2. Document Reference Number: CERN-LHCb-2005-073</b>			
<b>3. Issue</b>	<b>4. Revision</b>	<b>5. Date</b>	<b>6. Reason for change</b>
Draft	0	August 26, 2005	Initial version using the LHCb template.
Draft	1	September 5, 2005	Added PVSS section.
Final	0	September 12, 2005	Various small changes.
Final	1	September 28, 2005	Final revision.

## Contents

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Server	3
1.2	Command-line Client	3
<b>2</b>	<b>Configuration</b>	<b>4</b>
2.1	[global]-Section	4
2.2	[LHCb]-Section	4
<b>3</b>	<b>ELOG API</b>	<b>4</b>
3.1	Entry Submission (Command-line Client)	4
3.2	Display Search Results	4
<b>4</b>	<b>Adaptions</b>	<b>5</b>
4.1	Automatic Insertion of Fill- and Run-Number	5
4.2	Prefilled Author Name	5
<b>5</b>	<b>PVSS: fwLog Component</b>	<b>6</b>
5.1	Installation	6
5.2	Usage	6
5.3	Internal Details	6
<b>6</b>	<b>References</b>	<b>7</b>

## Introduction

ELOG will be used as the electronic logbook for the control room of the LHCb experiment. Its purpose is to provide the shift crew and internal or external experts with a quick and easy-to-use logbook to support their work. Figure 1 shows a screenshot of the ELOG start page.

The following requirements were specified by the LHCb Online Group and Olivier Callot:

- It has to be simple and user-friendly.
- A Web browser should be used to view, add and search logbook entries.
- Not only text entries but also pictures, hyperlinks and binary data should be supported.
- The logbook should add an automatic timestamp to new messages.
- There should be a possibility to preset attributes like the fill/run-number and the author name.

A very useful source not only while specifying the requirements was the ALEPH logbook [5] which was made by Olivier Callot. The link to ELOG was found at the ATLAS collaboration which used the software for a testbeam.

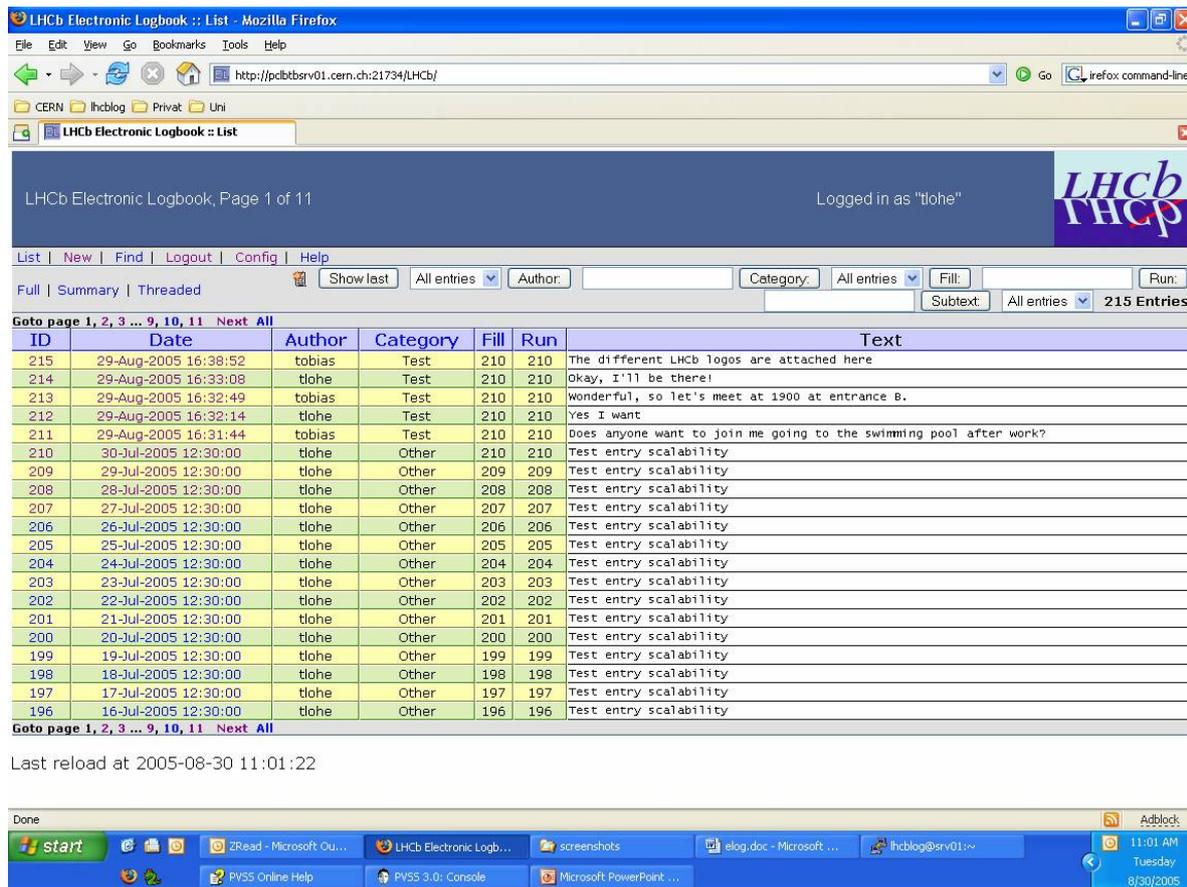


Figure 1: The ELOG start page.

The first section describes the installation procedure – including the copying of all additional scripts and files – on a Linux machine. The installation assumes that a preconfigured configuration file `elogd.cfg` already exists. It also explains how to install the command-line client for the submission of entries.

The second section describes the configuration file `elogd.cfg`.

The third section describes two APIs of ELOG, one for submitting entries via the command-line and a second to easily display or retrieve search results.

Section 4 deals with the additional scripts written for the adaption of ELOG.

In section 5 the PVSS framework component fwLog is described.

## 1 Installation

### 1.1 Server

The following steps are needed for the correct installation of the ELOG server:

1. Download and compile ELOG following the instructions on <http://midas.psi.ch/elog/download.html>.
2. Download the logo file `lhcblogo.gif`, the Python script `fillRunRegister.py` and the prepared configuration file `elogd.cfg` from the LHCb Web site [1] and copy them to the installation directory.
3. Modify the configuration file (`elogd.cfg`), especially regarding the URL, port, Usr, Grp, Logbook dir, and Admin user options in the [global]-Section and the Execute new-option in the [LHCb]-section of the configuration file. For detailed descriptions of these options see section 2.
4. Start ELOG as daemon using the command

```
elogd -c /path/to/elogd.cfg -D -x
```
5. The administrator should then try to connect to the logbook. If this is successful he should register with the admin user name given in the configuration file (via the Login-command and the Register as new user-button). This administrator account should in any case be protected with a password.

### 1.2 Command-line Client

The command-line client is provided in two versions, one for Linux and one for Windows.

#### *Linux*

1. Download the executable file `elog` from the LHCb Web site [1] and copy it to a convenient place on your machine.
2. Change permissions of `elog` using `chmod a+x elog`.
3. The usage of the command-line client is described in section 3.1.

#### *Windows*

1. Download the executable file `elog.exe` from the LHCb Web site [1] and copy it to a convenient place on your machine.
2. The usage of the command-line client is described in section 3.1.

## 2 Configuration

### 2.1 [global]-Section

The [global]-Section controls the settings for all logbooks, i.e. it mainly configures the behaviour of the `eologd` daemon.

The provided configuration file `eologd.cfg` is documented and almost self-explanatory. For option details see [3].

Especially important settings are the `port` and `URL`. As the logbook is running on the `srv01` inside the `daq.lhcb-domain` the `URL` has to be specified as the outside address `http://pclbtbsrv01.cern.ch`. Unfortunately the logbook cannot be accessed from inside the `daq.lhcb-domain` using this configuration. This is due to the fact that inside this domain the `URL` of the `srv01` is `srv01.daq.lhcb`. As ELOG uses the absolute `URL` given in the configuration file to redirect e.g. to new entries, any request from inside the `daq.lhcb-domain` will be redirected to the outside `URL` which is not accessible from within the domain. Solutions to this problem are either to mirror the logbook or to open a `SSH` connection to the gateway and to connect to the logbook from there.

It is also important to specify the *absolute* path to the logbook data (`Logbook dir-option`) if ELOG is run in daemon mode.

### 2.2 [LHCb]-Section

The [LHCb]-Section defines the behaviour of the LHCb logbook, like the menu commands, the quick filters, attributes and various display options.

Again, the provided configuration file `eologd.cfg` is documented and almost self-explanatory. For option details see [3].

Most of the options are straightforward, but it is very important to specify the absolute path names in the `Execute new-option` if ELOG is run in daemon mode.

## 3 ELOG API

### 3.1 Entry Submission (Command-line Client)

The command-line client program `eolog` can be used both on Linux and Windows in order to submit entries. It has to be installed on the local machine (which is described in section 1.2).

The syntax is the following ([ ] means optional, | alternative, \* 0...n times):

```
eolog -h <hostname> -p <port> -l lhcb
-a Author=<author>
-a Category=<category>
[-a Fill=<fill>]
[-a Run=<run>]
[-f attachment]*
"entry text" | -m textfile
```

### 3.2 Display Search Results

To display search results query strings (see Table 1) can be used. They have to be added to the `URL` of the LHCb logbook (`http://hostname:port/LHCb/`).

Two or more query strings can be combined using the `&` sign, e.g. `?last=1&Author=tlohe` displays all entries of the last day written by `tlohe`.

last day	?last=1
last week	?last=7
last x days	?last=x
Author=<name>	?Author=<name>
Category=<name>	?Category=<name>
Fill=<number>	?Fill=<number>
Run=<number>	?Run=<number>
full text search for <text>	?subtext=<text>

Table 1 Query Strings

## 4 Adaptions

There were two requirements ELOG could not fulfill out-of-the-box. First, the fill- and the run-number should be automatically inserted, so that the users don't have to struggle with finding them out. Secondly, the author name should be remembered and preset in the form for new entries, so that the users don't need to retype their names all the time.

### 4.1 Automatic Insertion of Fill- and Run-Number

For this requirement two assumptions were made:

1. The fill- and run-number remain stable for a given time.
2. If they change PVSS will submit a new entry containing the new fill- and run-number.

Based on this assumptions, the `Execute new`-option in the configuration file was used to call a Python script which after submission of the entry modifies the data files and writes the last known fill/run-combination to these files.

As the `Execute new`-mechanism was also used for the prefilled author name functionality covered in section 4.2 the Python script was named `fillRunRegister.py` and called from ELOG using the following line in `elogd.cfg` (the absolute path names were omitted but are necessary if ELOG is run as daemon)

```
Execute new = python fillRunRegister.py $message id $Author $Fill $Run &
```

which means that the message id, the author name, the fill- and run-number of the last entry are submitted as command-line arguments to the script. And the `&` tells Linux to run the command in the background.

The relevant function in `fillRunRegister.py` is called `writeFillRun` and uses the function `getFillRun`. Basically, the script extracts the current fill/run-pair from the data files and writes this pair to the last submitted entry. For details see the file `fillRunRegister.py`, which is documented and self-explanatory.

### 4.2 Prefilled Author Name

In this case the requirements were the following:

- The author name should be preset when a user wants to submit a new entry.
- But the users should not be bothered with registration, passwords and the need to login every time they connect to the logbook.

I exploited the built-in password mechanism of ELOG for this purpose, but with empty passwords so that no one has to remember passwords but can still be remembered by ELOG. To make it still easier for the users they are automatically registered if a new entry is made with an unknown author name.

Basically the registration is done by requesting the HTML form of ELOG to register a new user. The URL of the form is

```
http://hostname:port/LHCb/?cmd=Save&new_user_name=<name>&new_full_name=<name>.
```

This form is requested from the script `fillRunRegister.py` upon the submission of a new entry (using the `Execute new`-option in the configuration file).

For details see again the script `fillRunRegister.py`, which is documented and self-explanatory.

## 5 PVSS: fwLog Component

### 5.1 Installation

You can download the fwLog component from [4]. For the correct installation please follow the guidelines on this page (i.e. use the `fwInstallation` tool to install fwLog). During the installation process you will be asked if you want to add the `Fill/Run Control Manager` (see figure 2). Please answer with `Activate`, otherwise the whole component will not work properly.

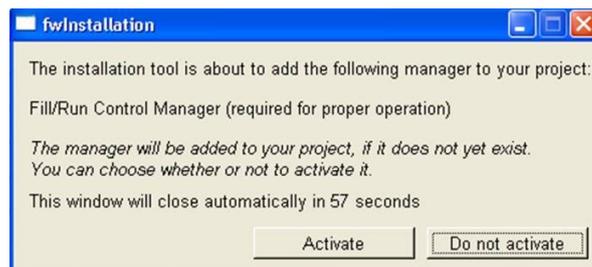


Figure 2: User interface asking whether to activate the control manager.

After the installation has finished the project will restart and you will find a new control manager added for the `fwLogFillRun.ctl`-script. Please make sure that this control manager is in `always start` mode and running, otherwise the component will not work properly.

You may have to adapt the `logbook datapoint` of `DPT LogbookInfo` to the correct server and port number ELOG is running.

### 5.2 Usage

You find the main panel under `panels/fwLog/fwLog.pnl`. Using this panel you can control every logbook relevant function (see figure 3), namely change the fill or run number, submit a new entry or display search results in the default browser (which you have to set on Linux).

### 5.3 Internal Details

The fwLog component includes several files:

- The ELOG command line client, either for Linux (`eelog`) or Windows (`eelog.exe`).
- The datapoint types and datapoints for the fill/run-number and the meta-information about the logbook (server, port, etc.)
- The user interface panels mentioned above.

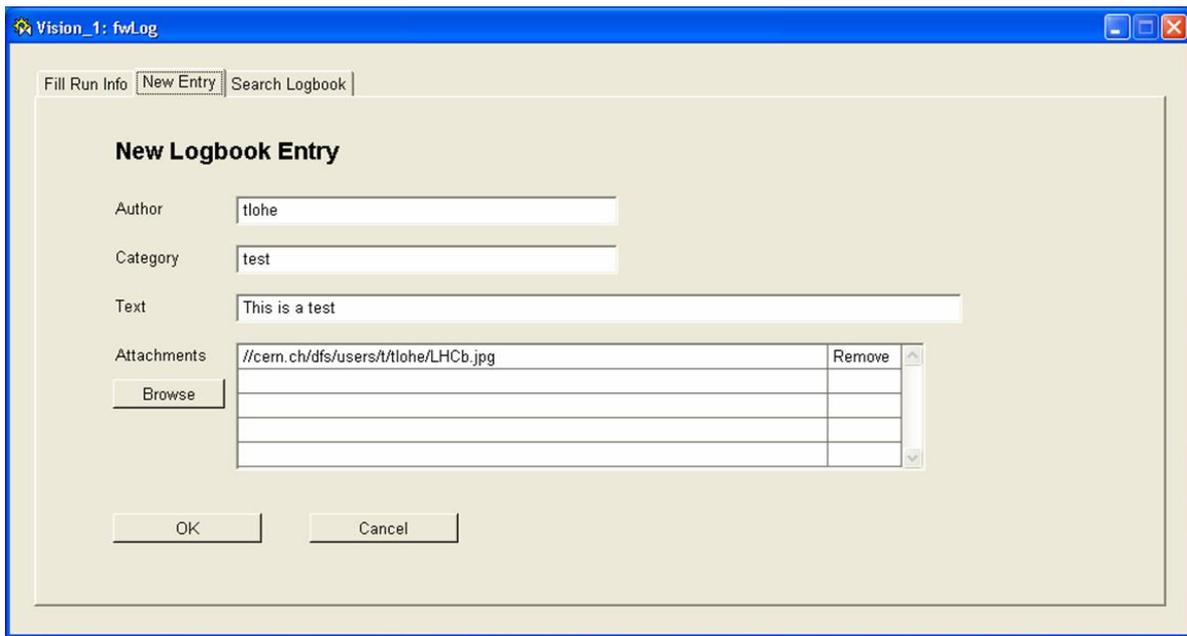


Figure 3: The main fwLog panel.

— The control script which supervises the fill/run-number and submits changes to the logbook.

It is important to adapt the logbook meta-information stored in the `logbook` datapoint of DPT `LogbookInfo` if the server or port change.

If the data model of the logbook (number and type of attributes) changes, all panels and the control script need to be adapted too.

## 6 References

- [1] LHCb Electronic Logbook Help. <http://lhcb-online.web.cern.ch/lhcb-online/elog/>
- [2] ELOG Home Page. <http://midas.psi.ch/elog/>
- [3] ELOG Configuration. <http://midas.psi.ch/elog/config.html>
- [4] FwLog PVSS Framework Component. <http://lhcb-online.web.cern.ch/lhcb-online/elog/fwLog1.0.zip>
- [5] ALEPH Online Logbook. <http://axaon1.cern.ch:8000/htbin/logbook/>