

WebGUI Logging

JT Smith :: WUC 2007

Logging?

WTF can you do with a log file?

log4perl

Log::Log4perl lets you remote-control and fine-tune the logging behaviour of your system from the outside. It implements the widely popular (Java-based) Log4j logging package in pure Perl.

log.conf

Logging is all about the config file.

log.conf

```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

log.conf

```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Category Name

In this case, the master category.

log.conf

```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Log Level

DEBUG, INFO, WARN, ERROR, FATAL

log.conf

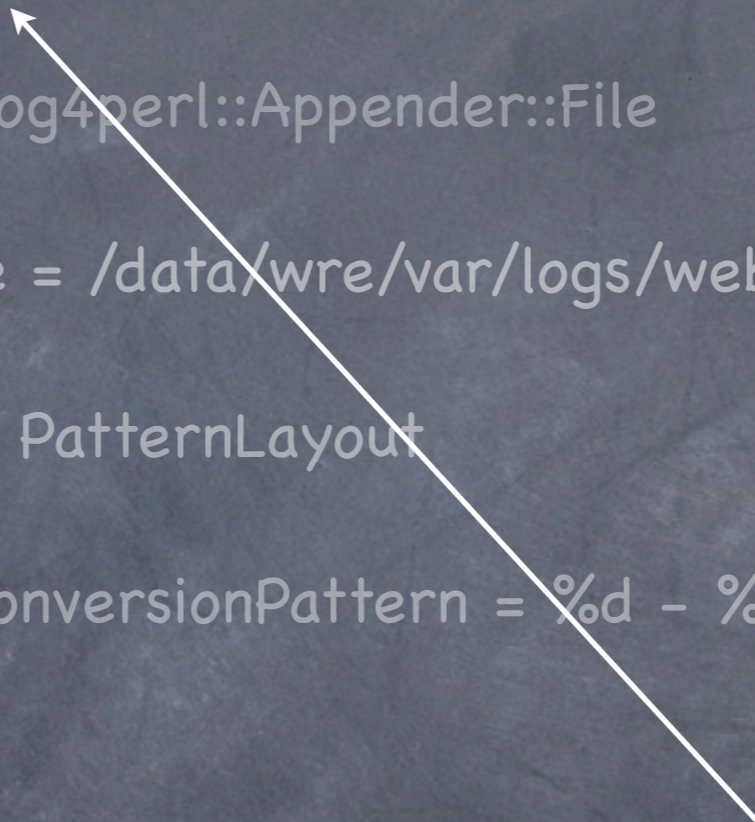
```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```



Log Alias
Used

Log Alias Defined

log.conf

```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Define a storage/display
mechanism called an
Appender



log.conf

```
log4perl.logger = WARN, mainlog
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Appender Configuration



Log Per Site

```
log4perl.logger.www.example.com = WARN, web
log4perl.appender.web = Log::Log4perl::Appender::File
log4perl.appender.web.filename = /data/wre/var/logs/www.example.com.log
log4perl.appender.web.layout = PatternLayout
log4perl.appender.web.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n

log4perl.logger.intranet.example.com = WARN, intranet
log4perl.appender.intranet = Log::Log4perl::Appender::File
log4perl.appender.intranet.filename = /data/wre/var/logs/intranet.example.com.log
log4perl.appender.intranet.layout = PatternLayout
log4perl.appender.intranet.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Log Per Site

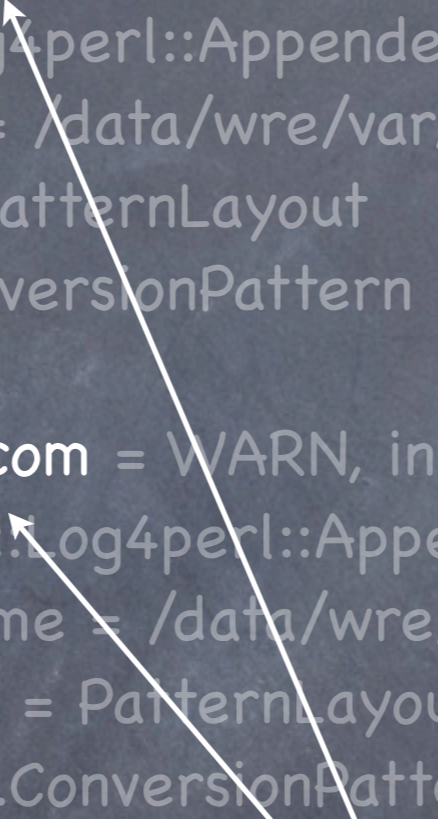
```
log4perl.logger.www.example.com = WARN, web
log4perl.appender.web = Log::Log4perl::Appender::File
log4perl.appender.web.filename = /data/wre/var/logs/www.example.com.log
log4perl.appender.web.layout = PatternLayout
log4perl.appender.web.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n

log4perl.logger.intranet.example.com = WARN, intranet
log4perl.appender.intranet = Log::Log4perl::Appender::File
log4perl.appender.intranet.filename = /data/wre/var/logs/intranet.example.com.log
log4perl.appender.intranet.layout = PatternLayout
log4perl.appender.intranet.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Log Per Site

```
log4perl.logger.www.example.com = WARN, web
log4perl.appender.web = Log::Log4perl::Appender::File
log4perl.appender.web.filename = /data/wre/var/logs/www.example.com.log
log4perl.appender.web.layout = PatternLayout
log4perl.appender.web.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n

log4perl.logger.intranet.example.com = WARN, intranet
log4perl.appender.intranet = Log::Log4perl::Appender::File
log4perl.appender.intranet.filename = /data/wre/var/logs/intranet.example.com.log
log4perl.appender.intranet.layout = PatternLayout
log4perl.appender.intranet.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```



WebGUI's category is the config file name.

Log Per Site

```
log4perl.logger.www.example = WARN, web  
log4perl.logger.www.example.com = WARN, web  
log4perl.logger.www.example.com.conf = WARN, web
```

Probably the same thing unless there is an `example.net` or `example.org` in your sites list.

Log Per Site

```
log4perl.logger.www = WARN, web
```

All sites that begin with www.

Log Per Site

`log4perl.logger.spectre.conf = WARN, web`

Anything that S.P.E.C.T.R.E. logs.

Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch
```

```
log4perl.MatchFatals.LevelToMatch = FATAL
```

```
log4perl.filter.MatchFatals.AcceptOnMatch = true
```

```
log4perl.appender.fatals.Filter = MatchFatals
```

```
log4perl.appender.fatals = Log::Log4perl::Appender::File
```

```
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log
```

```
log4perl.appender.fatals.layout = PatternLayout
```

```
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
log4perl.appender.fatals.Threshold = FATAL
```

Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch
```

```
log4perl.MatchFatals.LevelToMatch = FATAL
```

```
log4perl.filter.MatchFatals.AcceptOnMatch = true
```

```
log4perl.appender.fatals.Filter = MatchFatals
```

```
log4perl.appender.fatals = Log::Log4perl::Appender::File
```

```
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log
```

```
log4perl.appender.fatals.layout = PatternLayout
```

```
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
log4perl.appender.fatals.Threshold = FATAL
```

Define an additional
log alias

Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch
```

```
log4perl.MatchFatals.LevelToMatch = FATAL
```

```
log4perl.filter.MatchFatals.AcceptOnMatch = true
```

```
log4perl.appender.fatals.Filter = MatchFatals
```

```
log4perl.appender.fatals = Log::Log4perl::Appender::File
```

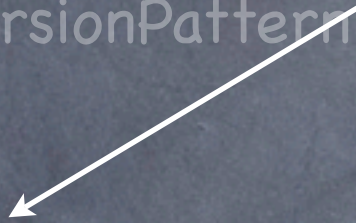
```
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log
```

```
log4perl.appender.fatals.layout = PatternLayout
```

```
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
log4perl.appender.fatals.Threshold = FATAL
```

Define a filter
that matches
only FATAL
messages.



Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File  
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log  
log4perl.appender.mainlog.layout = PatternLayout  
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Assign the filter
to our fatals log
alias.

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch  
log4perl.MatchFatals.LevelToMatch = FATAL  
log4perl.filter.MatchFatals.AcceptOnMatch = true  
log4perl.appender.fatals.Filter = MatchFatals  
log4perl.appender.fatals = Log::Log4perl::Appender::File  
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log  
log4perl.appender.fatals.layout = PatternLayout  
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n  
log4perl.appender.fatals.Threshold = FATAL
```

Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Or instead of all that
we could just add this
line to our appender.

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch
```

```
log4perl.MatchFatals.LevelToMatch = FATAL
```

```
log4perl.filter.MatchFatals.AcceptOnMatch = true
```

```
log4perl.appender.fatals.Filter = MatchFatals
```

```
log4perl.appender.fatals = Log::Log4perl::Appender::File
```

```
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log
```

```
log4perl.appender.fatals.layout = PatternLayout
```

```
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

```
log4perl.appender.fatals.Threshold = FATAL
```

Fatals Log

```
log4perl.logger = WARN, mainlog, fatals
```

```
# main log file
```

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

```
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log
```

```
log4perl.appender.mainlog.layout = PatternLayout
```

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

The filter still has a place. You can use it for a specific level, where **Threshold will append everything below the specified level.**

```
# log only containing fatals
```

```
log4perl.MatchFatals = Log::Log4perl::Filter::LevelMatch
```

```
log4perl.MatchFatals.LevelToMatch = FATAL
```

```
log4perl.filter.MatchFatals.AcceptOnMatch = true
```

```
log4perl.appender.fatals.Filter = MatchFatals
```

```
log4perl.appender.fatals = Log::Log4perl::Appender::File
```

```
log4perl.appender.fatals.filename = /data/wre/var/logs/webgui.fatal.log
```

```
log4perl.appender.fatals.layout = PatternLayout
```

```
log4perl.appender.fatals.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

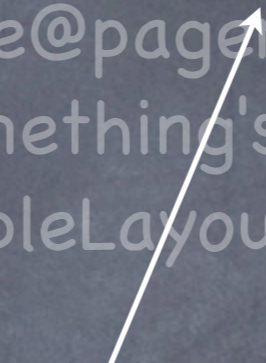
```
log4perl.appender.fatals.Threshold = FATAL
```

Email Yourself Fataals

```
log4perl.category = FATAL, Mailer
log4perl.appender.Mailer      = Log::Dispatch::Email::MailSend
log4perl.appender.Mailer.to    = drone@pageme.net
log4perl.appender.Mailer.subject = Something's broken!
log4perl.appender.Mailer.layout = SimpleLayout
log4perl.appender.Mailer.buffered = 0
```

Email Yourself Fataals

```
log4perl.category = FATAL, Mailer
log4perl.appender.Mailer = Log::Dispatch::Email::MailSend
log4perl.appender.Mailer.to = drone@pageme.net
log4perl.appender.Mailer.subject = Something's broken!
log4perl.appender.Mailer.layout = SimpleLayout
log4perl.appender.Mailer.buffered = 0
```



You need to install this appender, because it doesn't come with log4perl.

Email Yourself Fataals

```
log4perl.category = FATAL, Mailer
log4perl.appender.Mailer = Log::Dispatch::Email::MailSend
log4perl.appender.Mailer.to = drone@pageme.net
log4perl.appender.Mailer.subject = Something's broken!
log4perl.appender.Mailer.layout = SimpleLayout
log4perl.appender.Mailer.buffered = 0
```

Your email address. Or
better yet, someone else's
email address. =)

Email Yourself Fataals

```
log4perl.category = FATAL, Mailer
log4perl.appender.Mailer = Log::Dispatch::Email::MailSend
log4perl.appender.Mailer.to = drone@pageme.net
log4perl.appender.Mailer.subject = Something's broken!
log4perl.appender.Mailer.layout = SimpleLayout
log4perl.appender.Mailer.buffered = 0
```

A subject line. I recommend something unique that you can use email filters on. Putting hostname in subject also a good idea.

Email Yourself Fataals

```
log4perl.category = FATAL, Mailer
log4perl.appender.Mailer = Log::Dispatch::Email::MailSend
log4perl.appender.Mailer.to = drone@pageme.net
log4perl.appender.Mailer.subject = Something's broken!
log4perl.appender.Mailer.layout = SimpleLayout
log4perl.appender.Mailer.buffered = 0
```

Without this it would only
email after building up a
few messages.

IM Yourself FATALs

```
log4perl.category = FATAL, Jabber
log4perl.appender.Jabber = Log::Dispatch::Jabber
log4perl.appender.Jabber.login.hostname = jabber.example.com
log4perl.appender.Jabber.login.port = 5222
log4perl.appender.Jabber.login.username = logger
log4perl.appender.Jabber.login.password = 123qwe
log4perl.appender.Jabber.login.resource = logger
log4perl.appender.Jabber.to = drone@jabber.example.com
log4perl.appender.Jabber.layout = SimpleLayout
log4perl.appender.Jabber.buffer = 3
```

IM Yourself FATALs

```
log4perl.category = FATAL, Jabber
log4perl.appender.Jabber = Log::Dispatch::Jabber
log4perl.appender.Jabber.login.hostname = jabber.example.com
log4perl.appender.Jabber.login.port = 5222
log4perl.appender.Jabber.login.username = logger
log4perl.appender.Jabber.login.password = 123qwe
log4perl.appender.Jabber.login.resource = logger
log4perl.appender.Jabber.to = drone@jabber.example.com
log4perl.appender.Jabber.layout = SimpleLayout
log4perl.appender.Jabber.buffer = 3
```

You'll have to install
this module.

IM Yourself FATALs

```
log4perl.category = FATAL, Jabber
log4perl.appender.Jabber = Log::Dispatch::Jabber
log4perl.appender.Jabber.login.hostname = jabber.example.com
log4perl.appender.Jabber.login.port = 5222
log4perl.appender.Jabber.login.username = webgui
log4perl.appender.Jabber.login.password = 123qwe
log4perl.appender.Jabber.login.resource = logger
log4perl.appender.Jabber.to = drone@jabber.example.com
log4perl.appender.Jabber.layout = SimpleLayout
log4perl.appender.Jabber.buffer = 3
```

Specify all the login information required to access your jabber server. The logger needs it's own account.

IM Yourself Fataals

```
log4perl.category = FATAL, Jabber
log4perl.appender.Jabber = Log::Dispatch::Jabber
log4perl.appender.Jabber.login.hostname = jabber.example.com
log4perl.appender.Jabber.login.port = 5222
log4perl.appender.Jabber.login.username = logger
log4perl.appender.Jabber.login.password = 123qwe
log4perl.appender.Jabber.login.resource = logger
log4perl.appender.Jabber.to = drone@jabber.example.com
log4perl.appender.Jabber.layout = SimpleLayout
log4perl.appender.Jabber.buffer = 3
```

Specify who gets the message.

IM Yourself FATALs

```
log4perl.category = FATAL, Jabber
log4perl.appender.Jabber = Log::Dispatch::Jabber
log4perl.appender.Jabber.login.hostname = jabber.example.com
log4perl.appender.Jabber.login.port = 5222
log4perl.appender.Jabber.login.username = logger
log4perl.appender.Jabber.login.password = 123qwe
log4perl.appender.Jabber.login.resource = logger
log4perl.appender.Jabber.to = drone@jabber.example.com
log4perl.appender.Jabber.layout = SimpleLayout
log4perl.appender.Jabber.buffer = 3
```

Specify how many fatals
should be generated
before sending a message.

Log to XML

```
log4perl.logger = WARN, mainlog  
log4perl.appender.mainlog = Log::Log4perl::Appender::File  
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log  
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::XMLLayout  
log4perl.appender.mainlog.layout.LocationInfo = TRUE  
log4perl.appender.mainlog.layout.Encoding = iso8859-1
```

Note that you have to install
`Log::Log4perl::Layout::XMLLayout`
separately.

Log to XML

```
log4perl.logger = WARN, mainlog  
log4perl.appender.mainlog = Log::Log4perl::Appender::File  
log4perl.appender.mainlog.filename = /data/wre/var/logs/webgui.log  
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::XMLLayout  
log4perl.appender.mainlog.layout.LocationInfo = TRUE  
log4perl.appender.mainlog.layout.Encoding = iso8859-1
```

Just a new kind of layout system.
But it's easily readable by log file
analyzers like Chainsaw.

Chainsaw v2 - Log Viewer

File View Current tab Help

Refine focus on:

ID	Timestamp	Level	Logger	Thread	Message
142	2004-05-12 15:43:02,311	Info	com.mycompany....	Thread-1	infomsg 141
143	2004-05-12 15:43:02,311	Warn	com.mycompany....	Thread-1	warnmsg 142
144	2004-05-12 15:43:02,311	Error	com.someotherco...	Thread-1	errmsg 143
145	2004-05-12 15:43:03,313	Debug	com.mycompany....	Thread-1	debugmsg 144 g dg sc
146	2004-05-12 15:43:03,313	Info	com.mycompany....	Thread-1	infomsg 145
147	2004-05-12 15:43:03,313	Warn	com.someotherco...	Thread-1	warnmsg 146
148	2004-05-12 15:43:03,313	Error	com.mycompany....	Thread-1	errmsg 147
149	2004-05-12 15:43:03,313	Debug	com.mycompany....	Thread-1	debugmsg 148
150	2004-05-12 15:43:03,313	Info	com.someotherco...	Thread-1	infomsg 149
151	2004-05-12 15:43:03,313	Warn	com.mycompany....	Thread-1	warnmsg 150
152	2004-05-12 15:43:03,313	Error	com.mycompany....	Thread-1	errmsg 151
153	2004-05-12 15:43:03,313	Debug	com.someotherco...	Thread-1	debugmsg 152
154	2004-05-12 15:43:03,313	Info	com.mycompany....	Thread-1	infomsg 153
155	2004-05-12 15:43:03,313	Warn	com.mycompany....	Thread-1	warnmsg 154
156	2004-05-12 15:43:03,313	Error	com.someotherco...	Thread-1	errmsg 155

Level ERROR
 Logger com.someothercompany.corecomponent
 Time 2004-05-12 15:43:03,313
 Thread Thread-1
 Message errmsg 155
 NDC null
 Class
 Method
 Line
 File
 Properties {{hostname,localhost}}(some string,some valueGenerator 3){log4jid,156}{application,Generator 3}}
 Throwable org.apache.log4j.chainsaw.Generator.run(Unknown Source) at java.lang.Thread.run(Thread.java:534)

localhost-Generator 3 localhost-Generator 2 localhost-Generator 1 ChainsawCentral Welcome

Receiver's panel:false 0:0 0.0/s

Chainsaw

Chainsaw Tutorial

Start Tutorial Stop Tutorial

Welcome to the Chainsaw v2 Tutorial. Here you will learn how to effectively utilise the many features of Chainsaw.

[Expressions](#)

[Color filters](#)

[Display filters](#)

Conventions

To assist you, the following documentation conventions will be used

- Interesting items will be shown like this
- Things you should try during the tutorial will be shown like this

Outline

The built-in tutorial installs several "pretend" Receiver plugins that generate some example LoggingEvents and post them into Log4j just like a real Receiver.

- If you would like to read more about Receivers first, then click here. **(TODO)**

When you are ready to begin the tutorial, [click here](#), or click the "Start Tutorial" button in this dialog's toolbar.

Receivers

After you have said yes to the confirmation dialog, you should see 3 new tabs appear in the main GUI. This is because the tutorial has installed 3 'Generator' Receivers into the Log4j engine.

• Confirm this by checking the Receivers

Log to a Database

```
log4perl.logger = WARN, mainlog
log4perl.appender.mainlog = Log::Log4perl::Appender::DBI
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::PatternLayout
log4perl.appender.mainlog.datasource = DBI:mysql:thebigdb
log4perl.appender.mainlog.username = webgui
log4perl.appender.mainlog.password = 123qwe
log4perl.appender.mainlog.sql = insert into my_log_table \
    (loglevel, site, date, module, lineNumber, message) values (?, ?, ?, ?, ?, ?)
log4perl.appender.mainlog.params.1 = %p
log4perl.appender.mainlog.params.2 = %c
log4perl.appender.mainlog.params.3 = %d
log4perl.appender.mainlog.params.4 = %M
log4perl.appender.mainlog.params.5 = %L
log4perl.appender.mainlog.params.6 = %m
log4perl.appender.mainlog.bufferSize = 3
```

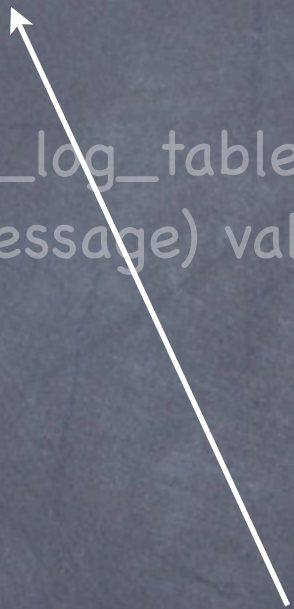
Log to a Database

```
log4perl.logger = WARN, mainlog
log4perl.appender.mainlog = Log::Log4perl::Appender::DBI
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::PatternLayout
log4perl.appender.mainlog.datasource = DBI:mysql:thebigdb
log4perl.appender.mainlog.username = webgui
log4perl.appender.mainlog.password = 123qwe
log4perl.appender.mainlog.sql = insert into my_log_table \
    (loglevel, site, date, module, lineNumber, message) values (?, ?, ?, ?, ?, ?)
log4perl.appender.mainlog.params.1 = %p
log4perl.appender.mainlog.params.2 = %c
log4perl.appender.mainlog.params.3 = %d
log4perl.appender.mainlog.params.4 = %M
log4perl.appender.mainlog.params.5 = %L
log4perl.appender.mainlog.params.6 = %m
log4perl.appender.mainlog.bufferSize = 3
```

We want to use the database appender.

Log to a Database

```
log4perl.logger = WARN, mainlog
log4perl.appender.mainlog = Log::Log4perl::Appender::DBI
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::PatternLayout
log4perl.appender.mainlog.datasource = DBI:mysql:thebigdb
log4perl.appender.mainlog.username = webgui
log4perl.appender.mainlog.password = 123qwe
log4perl.appender.mainlog.sql = insert into my_log_table \
    (loglevel, site, date, module, lineNumber, message) values (?, ?, ?, ?, ?, ?)
log4perl.appender.mainlog.params.1 = %p
log4perl.appender.mainlog.params.2 = %c
log4perl.appender.mainlog.params.3 = %d
log4perl.appender.mainlog.params.4 = %M
log4perl.appender.mainlog.params.5 = %L
log4perl.appender.mainlog.params.6 = %m
log4perl.appender.mainlog.bufferSize = 3
```




Provide the database connection info.

Log to a Database

```
log4perl.logger = WARN, mainlog
log4perl.appender.mainlog = Log::Log4perl::Appender::DBI
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::PatternLayout
log4perl.appender.mainlog.datasource = DBI:mysql:thebigdb
log4perl.appender.mainlog.username = webgui
log4perl.appender.mainlog.password = 123qwe
log4perl.appender.mainlog.sql = insert into my_log_table \
    (loglevel, site, date, module, lineNumber, message) values (?, ?, ?, ?, ?, ?)
log4perl.appender.mainlog.params.1 = %p
log4perl.appender.mainlog.params.2 = %c
log4perl.appender.mainlog.params.3 = %d
log4perl.appender.mainlog.params.4 = %M
log4perl.appender.mainlog.params.5 = %L
log4perl.appender.mainlog.params.6 = %m
log4perl.appender.mainlog.bufferSize = 3
```

Tie the log parameters
to fields in a database
table.



Log to a Database

```
log4perl.logger = WARN, mainlog
log4perl.appender.mainlog = Log::Log4perl::Appender::DBI
log4perl.appender.mainlog.layout = Log::Log4perl::Layout::PatternLayout
log4perl.appender.mainlog.datasource = DBI:mysql:thebigdb
log4perl.appender.mainlog.username = webgui
log4perl.appender.mainlog.password = 123qwe
log4perl.appender.mainlog.sql = insert into my_log_table \
    (loglevel, site, date, module, lineNumber, message) values (?, ?, ?, ?, ?, ?)
log4perl.appender.mainlog.params.1 = %p
log4perl.appender.mainlog.params.2 = %c
log4perl.appender.mainlog.params.3 = %d
log4perl.appender.mainlog.params.4 = %M
log4perl.appender.mainlog.params.5 = %L
log4perl.appender.mainlog.params.6 = %m
log4perl.appender.mainlog.bufferSize = 3
```

For efficiency sake
only write after we've
buffered 3 log
messages.

Performance

Performance

- The more you log, the slower it will be.

Performance

- The more you log, the slower it will be.
- WARN is the highest level you should use on a production site, and ERROR is recommended for busy sites.

Performance

- The more you log, the slower it will be.
- WARN is the highest level you should use on a production site, and ERROR is recommended for busy sites.
- Use a site specific log if you must log with more detail.

Write an SMS Appender

```
package Log::Log4perl::SmsAppender;
use strict;
use Net::SMS;

sub log {
    my $self = shift;
    my $sms = Net::SMS->new();
    $sms->subscriberID($self->{subscriberID});
    $sms->subscriberPassword($self->{subscriberPassword});
    $sms->msgPin($self->{to});
    $sms->msgFrom("WebGUI");
    $sms->msgCallback($self->{from});
    $sms->msgText($self->{message});
    $sms->optFlash($self->{gsmFlash} || "false");
    $sms->msgSend;
}

sub new {
    my ($class, %options) = @_;
    bless {%options}, $class;
}

1;
```

Write an SMS Appender

```
log4perl.category = FATAL, SMS
log4perl.appender.SMS = Log::Log4perl::SmsAppender
log4perl.appender.SMS.subscriberID = 123-456-789-12345
log4perl.appender.SMS.subscriberPassword = 123qwe
log4perl.appender.SMS.to = +1 100 410 1234
log4perl.appender.SMS.from = +1 100 510 1212
log4perl.appender.SMS.gsmFlash = true
log4perl.appender.SMS.layout = SimpleLayout
```

Write an SMS Appender

```
log4perl.category = FATAL, SMS
log4perl.appender.SMS = Log::Log4perl::SmsAppender
log4perl.appender.SMS.subscriberID = 123-456-789-12345
log4perl.appender.SMS.subscriberPassword = 123qwe
log4perl.appender.SMS.to = +1 100 410 1234
log4perl.appender.SMS.from = +1 100 510 1212
log4perl.appender.SMS.gsmFlash = true
log4perl.appender.SMS.layout = SimpleLayout
```

Write an SMS Appender

```
package Log::Log4perl::SmsAppender;
use strict;
use Net::SMS;

sub log {
    my $self = shift;
    my $sms = Net::SMS->new();
    $sms->subscriberID($self->{subscriberID});
    $sms->subscriberPassword($self->{subscriberPassword});
    $sms->msgPin($self->{to});
    $sms->msgFrom("WebGUI");
    $sms->msgCallback($self->{from});
    $sms->msgText($self->{message});
    $sms->optFlash($self->{gsmFlash} || "false");
    $sms->msgSend;
}

sub new {
    my ($class, %options) = @_;
    bless {%options}, $class;
}

1;
```

Write an SMS Appender

```
package Log::Log4perl::SmsAppender;
use strict;
use Net::SMS;

sub log {
    my $self = shift;
    my $sms = Net::SMS->new();
    $sms->subscriberID($self->{subscriberID});
    $sms->subscriberPassword($self->{subscriberPassword});
    $sms->msgPin($self->{to});
    $sms->msgFrom("WebGUI");
    $sms->msgCallback($self->{from});
    $sms->msgText($self->{message});
    $sms->optFlash($self->{gsmFlash} || "false");
    $sms->msgSend;
}

sub new {
    my ($class, %options) = @_;
    bless {%options}, $class;
}

1;
```

Questions?