



## Elluminate *Live!* Log Files

For Elluminate *Live!* Server Administrators, the Elluminate *Live!* Log files can help you determine what transpired in a given session. This document provides an overview of some of the keys log entries to look for. The elive.log file will be located on your Elluminate *Live!* server. By default, it will be housed on the /ElluminateLive/server6\_0/logs/ directory though it can be moved.

### Launch Log

The first example (See Example 1.1) shows the launch of the Elluminate *Live!* service on start up.

#### **Example 1.1**

```
2005.01.24 08:05:18 main.WrapperSimpleAppMain: DebugFlag.setEnabled: log.load=ON
2005.01.24 08:05:18 main.WrapperSimpleAppMain: DebugFlag.setEnabled: log.launch=ON
2005.01.24 08:05:18 main.WrapperSimpleAppMain: DebugFlag.setEnabled: log.access=ON
2005.01.24 08:05:18 Elluminate Live! Academic Edition (Elluminate Live! Demo License, 2 users)
2005.01.24 08:05:18 Elluminate Live! Scheduler 6.0
```

applicationSharing	6.0.3.92
attendance	6.0.1.4
audio	6.0.5.49
breakoutRooms	6.0.1.11
calculator	6.0.3.15
chair	6.0.4.18
directMessaging	6.0.4.38
framework	6.0.7.115
hand	6.0.4.26
multimedia	6.0.1.89
participantInfo	6.0.5.20
polling	6.0.2.10
quiz	6.0.1.48
recorder	6.0.2.23
startTime	6.0.3.2
video	6.0.3.43
webTour	6.0.3.19
whiteboard	6.0.6.51
Crimson	1.1.3
JDOM	b8

Date/Time Stamp – Provides the date and time of entry

Version Information – Provides the version information and license amount

Scheduler – Shows what JAR files that are loaded and their version. These will vary depending on which edition and version of Elluminate *Live!* you are using.

Note:

- The first three lines are debugging information that has been placed in the code. You may disregard this information.
- The fourth line provides the version and the license file.
- The fifth line through the second last provides the scheduler version, the JAR file names and JAR file versions.
- The last line is the copyright information.

## Template Files

The second example (See Example 1.2) shows a template file loading. You will need to move the tmpl file to the examples (or temporary) folder to make the changes. **DO NOT** make changes to the tmpl file in a live directory as it may disrupt the session associated with the file. Once you have made changes to the file, move it back to the live directory and you may either wait for the service to re-load the template or you may re-start the service which will force the reload of the templates.

### Example 1.2

```
2005.01.24 08:05:18 Loading template 'C:\Program
Files\ElluminateLive\server6_0\sessions\ELMDefault.tmpl'
2005.01.24 08:05:19 Loading template 'C:\Program Files\ElluminateLive\server6_0\sessions\Default.tmpl'
```

Template Load:

- Date/time of load
- Template location and name

Whenever you make changes to these template files, you should check the log to ensure that no errors were thrown. If there was an error with the loading of a specific template file, you will see the line number of the template file that is causing the problem.

When launching into a room you may see a “406 error – unexpected response from server applet” – this may be due to the tmpl file either not being present, changes that have not been picked up, etc. You may need to either re-load the template file or re-start the service.

## Launching a Session

Two steps occur when a session is launched:

1. The session will launch (if not already running – See Example 1.3) and the user will be logged into the session.
2. The user will be logged into the newly launched session.

### **Example 1.3**

```
2005.01.21 12:05:45 Session Scheduler: Launching unscheduled session 'Meeting|meeting1' (Meeting|meeting1)
2005.01.21 12:05:45 Meeting|meeting1: Access GRANTED - participant//225.13.117.118
```

Launching a session:

- Date/time of session launch
- Session name
- Access Granted for each participants and their user name, IP address.

## Shutting Down a Session

When the last person logs out of a session the server will “Hold” the room in state for 5 minutes (or other, if re-defined in the grace period in the tmpl file). After that, the scheduler will shut down the session (See Example 1.4).

### **Example 1.4**

```
2005.01.21 14:45:29 Shutting down unscheduled session 'Meeting|meeting1' (Meeting|meeting1)
```

Shutting down a session:

- Date/time of session shutdown
- Session name

## Adding Information to the Log File

In the `elive.com` file (found at `/ElluminateLive/server6_0/` - See Example 1.5) there are parameters set for the log file output. These parameters that can be added to get a better picture of what is happening in a session.

### Example 1.5

```
<log file="C:\Program Files\ElluminateLive\server6_0\log\elive.log" size="250000">
  log.load
  log.launch
  log.access
</log>
```

There are quite a number of messages that can be added if you need more information. These can be added to the log section of the `elive.conf` file. Here are the most common:

- |                           |   |
|---------------------------|---|
| <b>log.memory</b>         | Causes the server to issue JVM heap memory consumption reports approximately once per hour  |
| <b>jinx.hangup</b>        | Causes the server to issue messages when clients are intentionally disconnected from the server either by leaving the room, being forced out by a moderator, pushed out via the admin console tool, server shutdown, etc. |
| <b>jinx.disconnection</b> | Causes the server to issue messages when clients disconnect from the because of I/O problems (timed out on read, network link dropped, etc.)  |



Any changes done to the conf file requires a restart of the Elluminate *Live!* Server service



If you need additional information included in the file, please contact Elluminate's Technical Services Team at [support@lluminate.com](mailto:support@lluminate.com) to determine if there are additional parameters that can be set.

---

## Interpreting Disconnect Logs

There are a few reasons for a participant exiting a session. Also, there are a few different types of disconnects from a session. The most common is a “remote disconnect” (See Example 1.6) which means the user disconnected themselves from the session.

### Example 1.6

```
2005.01.21 14:40:29 Meeting|meeting1.netRecv('participant',6): NetworkTransceiver.run: User participant(/225.13.117.118) has been disconnected remotely.
```

Disconnect information:

- User date/time of disconnect
- Session name
- Code Call (NetworkTransceiver.run:)
- User name
- User IP address
- Reason for disconnect

Disconnect reasons:

- **User has been disconnected remotely:**
  - User disconnected manually from the session or lost their Internet connection.
- **Input exception - java.net.SocketException: Connection reset**
  - This is the result of a TCP connection RESET request. It most likely came from the client system, proxy, firewall or router in the path from the client system to the server. This could be as the result of a network error seen by the client system causing its TCP/IP stack to believe the connection is no longer capable of operation and thus resetting the connection.
- **Input exception - java.io.EOFExcept**
  - The network connection has likely been dropped and pending input (read) operation for that client then takes this exception. This not an error in the software, just how the disconnection is seen through the Java network and I/O services.
- **Input exception - java.net.SocketTimeoutException: Read timed out**
  - No watch-dog I/O was received from the client system prior to the watch-dog timer expiring. The connection was dropped by the server as it is believed the client is no longer responding. This could be caused by network disconnects or the client system hanging, looping, crashing, be rebooted. If the client dismisses the application on the desktop system, that would normally be seen by the server and would be listed as a "has been disconnected remotely" log entry.
- **User has been disconnected locally (command)**
  - Either the session shutdown and the client was removed as part of shutdown, or the participant was ejected from the room.