MAUS - Bug #811

Stack trace in maus web app

02 December 2011 14:04 - Rogers, Chris

Status:	Rejected		**Start date:**	02 December 2011
Priority:	Normal		**Due date:**	
Assignee:	Jackson, Mike		**% Done:**	0%
Category:	Online reconstruction		**Estimated time:**	0.00 hour
Target version:	Future MAUS release			
Workflow:	In Development			

Description
I'm getting a stack trace in the maus web app output. I attach a log file. Any idea what is going on? Doesn't appear to have much affect on running, but would like to know whether there is any problem...

History

#1 - 02 December 2011 14:06 - Rogers, Chris

Should say - this is maus-app started in the usual way, like

```
python src/maus-web/manage.py runserver localhost:9000
```

Probably difficult to debug at distance, but just so you are aware of the issue...

#2 - 02 December 2011 14:07 - Tunnell, Christopher

That may have been because the web client left the website before thumbnails were generated.

Out of curiosity, I created 100 fake .eps files just to see what would happen and those errors would generate if I left the webpage. Do you get it in normal running (keeping in mind this is a prototype)?

#3 - 02 December 2011 14:08 - Tunnell, Christopher

And as Chris Rogers says, we're only running the demo server until we setup something more 'professional'/'secure'

#4 - 02 December 2011 14:11 - Rogers, Chris

I left it running over lunch and haven't been back down to the control room since like an hour ago. I just phoned Yordan - he was on the other onrec machine, so the only activity probably was me logging in by ssh from my office (just before making the log file).

#5 - 02 December 2011 14:12 - Rogers, Chris

I should say, just before scp'ing out the log file - that has been taking data over lunch.

#6 - 02 December 2011 14:40 - Jackson, Mike

OK, I've occasionally seen that one too, usually when getting impatient and hitting refresh too many times. It's a Django-web browser "quirk" (partly a side-effect of using their "development server" instead of a proper web server. It occurs when the browser breaks the connection while the server is still sending data.

There's a (closed) Django ticket about it at, https://code.djangoproject.com/ticket/4444, with the pertinent comment being:

The reason for this patch is not to 'fix' the broken pipe issue. As it's not really an issue. But there seems to be a lack of clear consensus on how serious this is.

According to many sources the 'Broken Pipe' is a normal browser quirk. For example, the browser reads from the socket and then decides that the image it's been reading pparently didn't change. The browser now this (forcefully) closes the connection ecause it does not need more data. The other end of this socket (the python runserver) now raises a socket exception telling the program that the client 'Broke the socket pipe'.

If this is really only a non-issue, aka browsers do this often and it's not serious, then i suggested that we at least change this to a slightly less scary looking error. The idea here is to turn an exception (which is programming lingo) into an user error.

#7 - 02 December 2011 14:45 - Rogers, Chris

I don't think that's it - I started the maus-app running, went away for lunch. Meanwhile, they used a kvm switch to look at another PC - so they weren't even working on the machine where this was running. Maybe it's caused by auto-refresh setting in the browser/web page? Probably not critical, may be something weird about the way we're running in the control room, thought you'd want to know about it...

#8 - 22 December 2011 17:24 - Jackson, Mike

Likely a race condition when reading to convert to thumbnail and the file is overwritten on the next iteration.

#9 - 22 December 2011 17:26 - Rogers, Chris

- Workflow set to In Development

Can we catch the exceptions then? Just makes our users feel happier if they can't see scary error messages (and when real error messages appear, they actually investigate rather than just ignoring).

#10 - 22 December 2011 17:28 - Tunnell, Christopher

Just to clarify: during normal running, people should only be looking at the webpage. This is for helping debug problems and not going down false paths. Right?

#11 - 22 December 2011 18:07 - Rogers, Chris

My feeling is that if it's visible in the control room, people will want to know whats going on... maybe just means we need to pipe stderr to a file (could be done overloading sys.stderr or whatever).

#12 - 22 December 2011 18:09 - Tunnell, Christopher

But this is a temporary webserver. Ideally we'll get our own monitor that will just cycle these plots. There will probably be a more interactive machine too but I doubt if it'll be the same machine that actually runs the webserver. Getting rid of the stack trace is good if it takes less than an hour or so... think of the MongoDB stuff: you can find the logs if you want, but if it threw a warning you wouldn't know as a user.

#13 - 22 December 2011 18:11 - Rogers, Chris

Fair enough...

#14 - 22 December 2011 19:04 - Jackson, Mike

It's an issue with Django and they won't fix it. The bug https://code.djangoproject.com/ticket/4444 was reported years ago and is "closed:wontfix". Subsequent reports have just been marked as duplicates of that. They blame it on browser's closing connections prematurely (fair enough) but requests for a more helpful error message seemed to be ignored.

I tried adding try/except blocks to our Django code but the except blocks aren't entered when the exception occurs. In a dedicated web server it would be buried in the log file.

I tried redirecting sys.stderr to a file (editing src/maus-web/manage.py). The console output becomes:


```
Validating models...

0 errors found
Django version 1.3.1, using settings 'maus-web.settings'
Development server is running at http://maus.epcc.ed.ac.uk:9000/
Quit the server with CONTROL-C.
--------------------------------------
Exception happened during processing of request from ('129.215.62.218', 55179)
--------------------------------------
--------------------------------------
Exception happened during processing of request from ('129.215.62.218', 55184)
--------------------------------------
```

Redirecting both stdout and stderr to files results in a silent console but the stdout file lacks any content.

The control room monitor team should only be concerned if:

- There is no web page.
- There are missing images.
- The refresh time stamp hasn't changed for a while.

So, a simple solution would be for them to minimise the console window after starting it.

#15 - 22 December 2011 19:22 - Tunnell, Christopher

- Status changed from Open to Rejected

Not a real problem. Will not even be a false alarm once we switch to apache.

#16 - 22 December 2011 20:04 - Rogers, Chris

Good stuff

#17 - 13 February 2012 09:22 - Jackson, Mike

Stack trace does not arise in Apache 2.2 usage.

Files

log_file	392 KB 02 December 2011	Rogers, Chris