

## openQA Project - action #93342

### Exception in log parser should Incomplete if the parser can't parse or throws an exception

2021-06-01 09:24 - cdywan

<b>Status:</b>	Workable	<b>Start date:</b>	2021-06-01
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Concrete Bugs	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	future		
<b>Difficulty:</b>			
<b>Description</b>			
<b>User story</b>			
As a test reviewer I need to know that failures are test failures and not due to internal parsing errors.			
<b>Acceptance criteria</b>			
<ul style="list-style-type: none"><li>• <b>AC1:</b> Failures in JSON results parsing don't lead to failed tests</li><li>• <b>AC2:</b> Tests incomplete if parsing is broken</li></ul>			
<b>Suggestions</b>			
<ul style="list-style-type: none"><li>• Check if JSON contains all expected fields e.g. validate against a schema and Incomplete</li><li>• Consider wrapping the parsing into an exception handler (try or eval) and Incomplete</li></ul>			
<b>Related issues:</b>			
Copied from openQA Project - action #93324: Exception in log parser leads to ...		<b>Workable</b>	<b>2021-06-01</b>

## History

### #1 - 2021-06-01 09:24 - cdywan

- Copied from action #93324: Exception in log parser leads to failed job with hard to guess root cause added

### #2 - 2021-06-01 10:29 - mkittler

The incomplete should also have a specific reason. We should then also ignore the incompletes within our monitoring because if a 3rdparty testsuite produces broken output (or uses an unexpected format) that's not really an issue on our side.

### #3 - 2021-06-01 10:33 - mkittler

Consider wrapping the parsing into an exception handler (try or eval) and Incomplete

Note that the parsing via the test API function `parse_extra_log` is already been wrapped into an exception handler. Maybe it isn't handled in the right way, though. Note that as far as I see none of the backtraces within the attached log are actually backtraces of exceptions (I see only warnings) so messing with the exception handling likely wouldn't make any difference here. As mentioned in [#93324#note-7](#) I would rather fix the code so warnings cannot occur anymore.

### #4 - 2021-06-01 10:41 - mkittler

- Assignee set to mkittler

### #5 - 2021-06-01 13:44 - mkittler

- File `screenshot_20210601_151810.png` added

- Status changed from Workable to Feedback

This is how it looks like *without* any modifications if an exception occurs within the parser code:

Results for **opensuse-Tumbleweed-DVD-x86\_64-Build20210527-btrfs@64bit** 0 ★ 1 ★

Result: **failed** finished 3 minutes ago ( 03:13 minutes ) 🔗

Clone of 2015

Scheduled product: job has not been created by posting an ISO (but possibly the original job)

Assigned worker: linux-9lfz:1

<https://progress.opensuse.org/issues/16570> PARTITIONING\_WARNINGS=1 Test installation with btrfs filesystem. Validates no\_COW attributes on subvolumes, checks the /home is not on the separate partition.

Details Logs & Assets Settings Investigation Comments (0) Next & previous results

Test	Result	References
Installation		
bootloader_start	failed	

```
# Test died: ':' expected, at character offset 12 (before "outcome": "passed", ...) at /usr/lib/perl5/vendor_perl/5.32.1/Mojo/JSON.pm line 33.
Mojo::JSON::from_json("{\"tests\": [{\"outcome\": \"passed\", \"test_index\": 0, \"nodeid\": \"t\"...}) called at /hdd/openqa-devel/repos/openQA/lib/OpenQA/Par
OpenQA::Parser::Format::IPA::parse(OpenQA::Parser::Format::IPA=HASH(0x5602342469f0), \"{\"tests\": [{\"outcome\": \"passed\", \"test_index\": 0, \"nodeid\": 0, \"nodeid\": \"t\"...}) called at /hdd/o
OpenQA::Parser::load(OpenQA::Parser::Format::IPA=HASH(0x5602342469f0), "/hdd/openqa-devel/openqa/share/tests/opensuse/img_proof-fault"...)) called at /hdd/o
OpenQA::Parser::parser("IPA", "/hdd/openqa-devel/openqa/share/tests/opensuse/img_proof-fault"...)) called at /hdd/openqa-devel/repos/os-autoinst/testapi.pm
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/testapi.pm line 2105
testapi::parse_extra_log("IPA", "/hdd/openqa-devel/openqa/share/tests/opensuse/img_proof-fault"...)) called at opensuse/tests/installation/bootloader_start.
bootloader_start::run(bootloader_start=HASH(0x560231cd3208)) called at /hdd/openqa-devel/repos/os-autoinst/basetest.pm line 356
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/basetest.pm line 356
basetest::runtest(bootloader_start=HASH(0x560231cd3208)) called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 374
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 374
autotest::runalltests() called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 242
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 242
autotest::run_all() called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 298
autotest::__ANON__(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70)) called at /usr/lib/perl5/vendor_perl/5.32.1/Mojo/IOLoop/ReadWriteProcess.pm line 32
eval {...} called at /usr/lib/perl5/vendor_perl/5.32.1/Mojo/IOLoop/ReadWriteProcess.pm line 326
Mojo::IOLoop::ReadWriteProcess::fork(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70), CODE(0x560233d544c8)) called at /usr/lib/perl5/vendor_perl/5.32.
Mojo::IOLoop::ReadWriteProcess::start(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70)) called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 3
autotest::start_process() called at /hdd/openqa-devel/repos/os-autoinst/isotovideo line 274
at /hdd/openqa-devel/repos/os-autoinst/testapi.pm line 2118.
testapi::parse_extra_log("IPA", "/hdd/openqa-devel/openqa/share/tests/opensuse/img_proof-fault"...)) called at opensuse/tests/installation/bootloader_start.
bootloader_start::run(bootloader_start=HASH(0x560231cd3208)) called at /hdd/openqa-devel/repos/os-autoinst/basetest.pm line 356
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/basetest.pm line 356
basetest::runtest(bootloader_start=HASH(0x560231cd3208)) called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 374
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 374
autotest::runalltests() called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 242
eval {...} called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 242
autotest::run_all() called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 298
autotest::__ANON__(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70)) called at /usr/lib/perl5/vendor_perl/5.32.1/Mojo/IOLoop/ReadWriteProcess.pm line 32
eval {...} called at /usr/lib/perl5/vendor_perl/5.32.1/Mojo/IOLoop/ReadWriteProcess.pm line 326
Mojo::IOLoop::ReadWriteProcess::fork(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70), CODE(0x560233d544c8)) called at /usr/lib/perl5/vendor_perl/5.32.
Mojo::IOLoop::ReadWriteProcess::start(Mojo::IOLoop::ReadWriteProcess=HASH(0x560233f46c70)) called at /hdd/openqa-devel/repos/os-autoinst/autotest.pm line 3
autotest::start_process() called at /hdd/openqa-devel/repos/os-autoinst/isotovideo line 274
```

Here the provided JSON file is broken and hence the JSON parser throws an exception. As stated in the previous comment, the log attached within the ticket description is a different case because there were only warnings. (Anton takes care of the warnings in <https://github.com/os-autoinst/openQA/pull/3926>.)

I'm wondering whether this way of handling exceptions within the parser code is actually bad. It is in fact in-line with other failing test API functions. Assuming a bug within the parser by default and not a problem with the data to be parsed seems wrong. But if there's just a problem with the data to be parsed that likely means tests need to be fixed or the parser needs to be adapted. That's similar to fixing tests or creating a new needle after a needle mismatch. In this case we also don't assume by default that os-autoinst's needle matching is broken.

Note that making this an incomplete (with a special reason) would likely also mean to stop the test execution there immediately instead of possibly continuing with the next test module.

So I'm wondering whether we should change this at all - especially since the ticket was mainly motivated by looking at test results where not even an exception has been thrown.

**#6 - 2021-06-01 14:15 - mkittrler**

- Status changed from *Feedback* to *Workable*
- Assignee deleted (mkittrler)

The ticket doesn't make much sense how I understand it. It would make sense to add a sanity check for the results the parser produced and produce a failure like in the screenshot above if the check fails.

**#7 - 2021-06-01 17:48 - cdywan**

The ticket doesn't make much sense how I understand it. It would make sense to add a sanity check for the results the parser produced and produce a failure like in the screenshot above if the check fails.

So including the other case you mentioned we have these scenarios:

- JSON is invalid, currently raises an exception
- JSON is missing expected fields, currently produces warnings

We don't seem to validate anything at the moment so there might even be more side effects. I think most importantly the behavior should be more well-defined. A modified JSON file shouldn't confuse the code. It should stop trying to process unknown results.

Note that making this an incomplete (with a special reason) would likely also mean to stop the test execution there immediately instead of possibly continuing with the next test module.

True. I'm a bit on the fence here. Stopping the test early seems like a good thing because *something* here is broken at that point. Maybe it should still *fail*, though?

**#8 - 2021-06-23 06:37 - okurz**

- Target version changed from *Ready* to *future*

**Files**

---

autoinst-log.txt	470 KB	2021-06-01	asmorodskyi
screenshot_20210601_151810.png	329 KB	2021-06-01	mkittler