

Database - Bug #6715

BufferManager.validate throws IllegalStateException if a validation fails via a ValidateException

08/26/2022 03:45 AM - Constantin Asofiei

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#2 - 08/26/2022 03:47 AM - Constantin Asofiei

If validation fails in BufferManager.validate, this abends:

```
catch (Exception e)
{
    throw new IllegalStateException("Exception at dirtyBuffers.flagIterating block", e);
}
```

with this exception:

```
Caused by: java.lang.IllegalStateException: Exception at dirtyBuffers.flagIterating block
at com.goldencode.p2j.persist.BufferManager.validate(BufferManager.java:1021)
at com.goldencode.p2j.util.TransactionManager.processValidate(TransactionManager.java:6959)
at com.goldencode.p2j.util.TransactionManager.access$10500(TransactionManager.java:658)
at com.goldencode.p2j.util.TransactionManager$TransactionHelper.processValidate(TransactionManager.java:9742)
at com.goldencode.p2j.util.BlockManager.lambda$processBody$0(BlockManager.java:8608)
at com.goldencode.p2j.util.BlockManager.executeStateManaged(BlockManager.java:11347)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:8608)
at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:9688)
at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:1467)
.....
```

```
Caused by: com.goldencode.p2j.util.ErrorConditionException: com.goldencode.p2j.persist.ValidationException: <record> already exists with <f1> "" <f2> "" <f3> ""
```

```
Caused by: com.goldencode.p2j.persist.ValidationException: <record> already exists with <f1> "" <f2> "" <f3> ""
```

#3 - 08/26/2022 10:56 AM - Greg Shah

Converting a legacy error into an abend seems like something easily avoided.

#4 - 10/14/2022 11:02 AM - Eric Faulhaber

Yes, `ErrorException` should be allowed through; it should not be wrapped in `IllegalStateException`. I think the catch block was added to handle Exception thrown by `AutoCloseable.close()`, but that's too broad of a net, and it doesn't help us to throw `IllegalStateException` from here; that's an exception that we probably should just log and eat. My bad, I missed this in code review. I'm surprised we haven't hit this sooner.

We could catch and rethrow `ConditionException` in another catch block inserted before the existing one, or (uglier) remove the catch block and manually close the `AutoCloseable` in a finally block (thus defeating the "auto" aspect).

Tijs, could you please check the files you committed related to the `TailMap` implementation? I just noticed that at least `TailMapAutoCloseable.java` has CRLF line endings, which is inconsistent with the rest of the source. I didn't check all the other files from that update.

#5 - 10/14/2022 11:07 AM - Constantin Asofiei

In some cases, a `ValidationException` (which is from a bug) is wrapped in a `ErrorException`, like this:

```
Caused by: com.goldencode.p2j.util.ErrorConditionException: com.goldencode.p2j.persist.ValidationException: Re
quested to persist DMO of type com.app.Tt_1_1__Impl__ with recid 14. A different DMO instance of type com.app.
Tt_1_1__Impl__ is already bound to this session with the same recid
```

I think we should distinguish from expected validation errors (like 'already exists') and these kind of abends.

#6 - 10/14/2022 11:14 AM - Eric Faulhaber

Constantin Asofiei wrote:

In some cases, a `ValidationException` (which is from a bug) is wrapped in a `ErrorException`, like this:
[...]

I think we should distinguish from expected validation errors (like 'already exists') and these kind of abends.

Yes, that is what my proposed approach of catching/rethrowing `ConditionException` would do. I don't think there is necessarily a real case for the Exception catch block, but the exception needs to be handled to satisfy the compiler, at least.

#7 - 10/14/2022 11:19 AM - Tijs Wickardt

Eric Faulhaber wrote:

Tijs, could you please check the files you committed related to the TailMap implementation? I just noticed that at least TailMapAutoCloseable.java has CRLF line endings, which is inconsistent with the rest of the source. I didn't check all the other files from that update.

That's not intentional of course. Thanks for spotting, I'll check all files and correct where needed.

#8 - 10/14/2022 03:04 PM - Tijs Wickardt

Tijs Wickardt wrote:

Eric Faulhaber wrote:

Tijs, could you please check the files you committed related to the TailMap implementation? I just noticed that at least TailMapAutoCloseable.java has CRLF line endings, which is inconsistent with the rest of the source. I didn't check all the other files from that update.

That's not intentional of course. Thanks for spotting, I'll check all files and correct where needed.

Eric, fixed in 3821c rev 14298..

I also did a quick scan of the src folder. The list below should be checked as well (not related to TailMap):

```
tw@lpt-tw-gcd:~/z/p2j$ find src -type f -name '*.java' -exec grep -qU $'\015' {} \; -ls
 77605258    32 -rw-rw-r--  1 tw      tw           31873 Oct 14 20:50 src/com/goldencode/p2j/util/longchar.j
ava
 78250283     8 -rw-rw-r--  1 tw      tw           7072 Oct 14 20:50 src/com/goldencode/p2j/oo/net/http/fil
ter/auth/BasicAuthenticationFilter.java
 78250594    16 -rw-rw-r--  1 tw      tw          15127 Oct 14 20:50 src/com/goldencode/p2j/oo/net/http/fil
ter/auth/DigestAuthenticationFilter.java
 78250960    20 -rw-rw-r--  1 tw      tw          16825 Oct 14 20:50 src/com/goldencode/p2j/oo/net/http/fil
ter/payload/JsonBodyWriter.java
 78250735    12 -rw-rw-r--  1 tw      tw           8939 Oct 14 20:50 src/com/goldencode/p2j/oo/net/http/fil
ter/payload/FormBodyWriter.java
 78250238    16 -rw-rw-r--  1 tw      tw          15924 Oct 14 20:50 src/com/goldencode/p2j/oo/net/http/Aut
henticatedRequest.java
```