Item 228
**git_comments:**

**git_commits:**

1. **summary:** KAFKA-6605: Fix NPE in Flatten when optional Struct is null (#5705)
   **message:** KAFKA-6605: Fix NPE in Flatten when optional Struct is null (#5705) Correct the Flatten SMT to properly handle null key or value `Struct` instances. Author: Michal Borowiecki <michal.borowiecki@openbet.com> Reviewers: Arjun Satish <arjun@confluent.io>, Robert Yokota <rayokota@gmail.com>, Randall Hauch <rhauch@gmail.com>

**github_issues:**

**github_issues_comments:**

**github_pulls:**

1. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)
2. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)
   **label:** documentation
3. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)
4. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)
5. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)
6. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist

(excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)

7. **title:** KAFKA-6605 fix NPE in Flatten when optional Struct is null
   **body:** *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes)

**github_pulls_comments:**

1. Hi @wicknicks , would you kindly be able to review this PR as well please?
2. **body:** @mihbor could you please update the description with a few more details. Particularly, the record/schema that caused the exception, and the inputs used to configure the transformation. thanks!
   **label:** documentation
3. @rhauch , can you merge this if it looks ok?
4. The jira ticket says it works from 2.3.1? is the timeline of 2.3.1 already planned? https://issues.apache.org/jira/browse/KAFKA-6605
5. @HungUnicorn doesn't look like there is a plan yet. Watch this page for updates: https://cwiki.apache.org/confluence/display/KAFKA/Future+release+plan

**github_pulls_reviews:**

**jira_issues:**

1. **summary:** Flatten SMT does not properly handle fields that are null
   **description:** When a message has a null field, the `Flatten` SMT does not properly handle this and throws an NPE. Consider this message from Debezium: {code} { "before": null, "after": { "dbserver1.mydb.team.Value": { "id": 1, "name": "kafka", "email": "kafka@apache.org", "last_modified": 1519939449000 } }, "source": { "version": { "string": "0.7.3" }, "name": "dbserver1", "server_id": 0, "ts_sec": 0, "gtid": null, "file": "mysql-bin.000003", "pos": 154, "row": 0, "snapshot": { "boolean": true }, "thread": null, "db": { "string": "mydb" }, "table": { "string": "team" } }, "op": "c", "ts_ms": { "long": 1519939520285 } } {code} Note how `before` is null; this event represents a row was INSERTED and thus there is no `before` state of the row. This results in an NPE: {noformat} java.lang.NullPointerException at org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:219) at org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:234) at org.apache.kafka.connect.transforms.Flatten.applyWithSchema(Flatten.java:151) at org.apache.kafka.connect.transforms.Flatten.apply(Flatten.java:75) at org.apache.kafka.connect.runtime.TransformationChain.apply(TransformationChain.java:38) at org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:211) at org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745) {noformat} Here's the connector configuration that was used: {code} { "name": "debezium-connector-flatten", "config": { "connector.class": "io.debezium.connector.mysql.MySqlConnector", "tasks.max": "1", "database.hostname": "mysql", "database.port": "3306", "database.user": "debezium", "database.password": "dbz", "database.server.id": "223345", "database.server.name": "dbserver-flatten", "database.whitelist": "mydb", "database.history.kafka.bootstrap.servers": "kafka-1:9092,kafka-2:9092,kafka-3:9092", "database.history.kafka.topic": "schema-flatten.mydb", "include.schema.changes": "true", "transforms": "flatten", "transforms.flatten.type": "org.apache.kafka.connect.transforms.Flatten$Value", "transforms.flatten.delimiter": "_" } } {code} Note that the above configuration sets the delimiter to `_`. The default delimiter is `.`, which is not a valid character within an Avro field, and doing this results in

the following exception: {noformat} org.apache.avro.SchemaParseException: Illegal character in: source.version at org.apache.avro.Schema.validateName(Schema.java:1151) at org.apache.avro.Schema.access$200(Schema.java:81) at org.apache.avro.Schema$Field.<init>(Schema.java:403) at org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2124) at org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2116) at org.apache.avro.SchemaBuilder$FieldBuilder.access$5300(SchemaBuilder.java:2034) at org.apache.avro.SchemaBuilder$GenericDefault.withDefault(SchemaBuilder.java:2423) at io.confluent.connect.avro.AvroData.addAvroRecordField(AvroData.java:898) at io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:799) at io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:652) at io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:647) at io.confluent.connect.avro.AvroData.fromConnectData(AvroData.java:324) at io.confluent.connect.avro.AvroConverter.fromConnectData(AvroConverter.java:75) at org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:220) at org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745) {noformat} This should probably be addressed in the documentation: when using Avro, set the delimiter to `_` or another alphanumeric character.

2. **summary:** Flatten SMT does not properly handle fields that are null
**description:** When a message has a null field, the `Flatten` SMT does not properly handle this and throws an NPE. Consider this message from Debezium: {code} { "before": null, "after": { "dbserver1.mydb.team.Value": { "id": 1, "name": "kafka", "email": "kafka@apache.org", "last_modified": 1519939449000 } }, "source": { "version": { "string": "0.7.3" }, "name": "dbserver1", "server_id": 0, "ts_sec": 0, "gtid": null, "file": "mysql-bin.000003", "pos": 154, "row": 0, "snapshot": { "boolean": true }, "thread": null, "db": { "string": "mydb" }, "table": { "string": "team" } }, "op": "c", "ts_ms": { "long": 1519939520285 } } {code} Note how `before` is null; this event represents a row was INSERTED and thus there is no `before` state of the row. This results in an NPE: {noformat} java.lang.NullPointerException at org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:219) at org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:234) at org.apache.kafka.connect.transforms.Flatten.applyWithSchema(Flatten.java:151) at org.apache.kafka.connect.transforms.Flatten.apply(Flatten.java:75) at org.apache.kafka.connect.runtime.TransformationChain.apply(TransformationChain.java:38) at org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:211) at org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745) {noformat} Here's the connector configuration that was used: {code} { "name": "debezium-connector-flatten", "config": { "connector.class": "io.debezium.connector.mysql.MySqlConnector", "tasks.max": "1", "database.hostname": "mysql", "database.port": "3306", "database.user": "debezium", "database.password": "dbz", "database.server.id": "223345", "database.server.name": "dbserver-flatten", "database.whitelist": "mydb", "database.history.kafka.bootstrap.servers": "kafka-1:9092,kafka-2:9092,kafka-3:9092", "database.history.kafka.topic": "schema-flatten.mydb", "include.schema.changes": "true", "transforms": "flatten", "transforms.flatten.type": "org.apache.kafka.connect.transforms.Flatten$Value", "transforms.flatten.delimiter": "_" } } {code} Note that the above configuration sets the delimiter to `_`. The default delimiter is `.`, which is not a valid character within an Avro field, and doing this results in the following exception: {noformat} org.apache.avro.SchemaParseException: Illegal character in: source.version at org.apache.avro.Schema.validateName(Schema.java:1151) at

org.apache.avro.Schema.access$200(Schema.java:81) at org.apache.avro.Schema$Field.<init>
(Schema.java:403) at
org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2124) at
org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2116) at
org.apache.avro.SchemaBuilder$FieldBuilder.access$5300(SchemaBuilder.java:2034) at
org.apache.avro.SchemaBuilder$GenericDefault.withDefault(SchemaBuilder.java:2423) at
io.confluent.connect.avro.AvroData.addAvroRecordField(AvroData.java:898) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:799) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:652) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:647) at
io.confluent.connect.avro.AvroData.fromConnectData(AvroData.java:324) at
io.confluent.connect.avro.AvroConverter.fromConnectData(AvroConverter.java:75) at
org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:220) at
org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at
org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at
org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) {noformat} This should probably be addressed in the documentation: when using Avro, set the delimiter to `_` or another alphanumeric character.

3. **summary:** Flatten SMT does not properly handle fields that are null
**description:** When a message has a null field, the `Flatten` SMT does not properly handle this and throws an NPE. Consider this message from Debezium: {code} { "before": null, "after": { "dbserver1.mydb.team.Value": { "id": 1, "name": "kafka", "email": "kafka@apache.org", "last_modified": 1519939449000 } }, "source": { "version": { "string": "0.7.3" }, "name": "dbserver1", "server_id": 0, "ts_sec": 0, "gtid": null, "file": "mysql-bin.000003", "pos": 154, "row": 0, "snapshot": { "boolean": true }, "thread": null, "db": { "string": "mydb" }, "table": { "string": "team" } }, "op": "c", "ts_ms": { "long": 1519939520285 } } {code} Note how `before` is null; this event represents a row was INSERTED and thus there is no `before` state of the row. This results in an NPE: {noformat}
java.lang.NullPointerException at
org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:219) at
org.apache.kafka.connect.transforms.Flatten.buildWithSchema(Flatten.java:234) at
org.apache.kafka.connect.transforms.Flatten.applyWithSchema(Flatten.java:151) at
org.apache.kafka.connect.transforms.Flatten.apply(Flatten.java:75) at
org.apache.kafka.connect.runtime.TransformationChain.apply(TransformationChain.java:38) at
org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:211) at
org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at
org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at
org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) {noformat} Here's the connector configuration that was used: {code} { "name": "debezium-connector-flatten", "config": { "connector.class": "io.debezium.connector.mysql.MySqlConnector", "tasks.max": "1", "database.hostname": "mysql", "database.port": "3306", "database.user": "debezium", "database.password": "dbz", "database.server.id": "223345", "database.server.name": "dbserver-flatten", "database.whitelist": "mydb", "database.history.kafka.bootstrap.servers": "kafka-1:9092,kafka-2:9092,kafka-3:9092", "database.history.kafka.topic": "schema-flatten.mydb", "include.schema.changes": "true", "transforms": "flatten", "transforms.flatten.type": "org.apache.kafka.connect.transforms.Flatten$Value", "transforms.flatten.delimiter": "_" } } {code} Note that the above configuration sets the delimiter to `_`. The default delimiter is `.`, which is not a valid character within an Avro field, and doing this results in the following exception: {noformat} org.apache.avro.SchemaParseException: Illegal character in: source.version at org.apache.avro.Schema.validateName(Schema.java:1151) at
org.apache.avro.Schema.access$200(Schema.java:81) at org.apache.avro.Schema$Field.<init>
(Schema.java:403) at

org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2124) at
org.apache.avro.SchemaBuilder$FieldBuilder.completeField(SchemaBuilder.java:2116) at
org.apache.avro.SchemaBuilder$FieldBuilder.access$5300(SchemaBuilder.java:2034) at
org.apache.avro.SchemaBuilder$GenericDefault.withDefault(SchemaBuilder.java:2423) at
io.confluent.connect.avro.AvroData.addAvroRecordField(AvroData.java:898) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:799) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:652) at
io.confluent.connect.avro.AvroData.fromConnectSchema(AvroData.java:647) at
io.confluent.connect.avro.AvroData.fromConnectData(AvroData.java:324) at
io.confluent.connect.avro.AvroConverter.fromConnectData(AvroConverter.java:75) at
org.apache.kafka.connect.runtime.WorkerSourceTask.sendRecords(WorkerSourceTask.java:220) at
org.apache.kafka.connect.runtime.WorkerSourceTask.execute(WorkerSourceTask.java:187) at
org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170) at
org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214) at
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) {noformat} This should probably be addressed in the documentation: when using Avro, set the delimiter to `_` or another alphanumeric character.

**jira_issues_comments:**

1. Excuse me for a second With recent support for nested data types does that present a suitable workaround for this issue? [https://www.confluent.io/blog/introducing-confluent-platform-5-0/] Just curious
2. mihbor opened a new pull request #5705: KAFKA-6605 fix NPE in Flatten when optional Struct is null URL: https://github.com/apache/kafka/pull/5705 *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes) ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
3. mihbor opened a new pull request #5706: KAFKA-6605 fix NPE in Flatten when optional Struct is null - backport… URL: https://github.com/apache/kafka/pull/5706 … to 2.0 *More detailed description of your change, if necessary. The PR title and PR message become the squashed commit message, so use a separate comment to ping reviewers.* *Summary of testing strategy (including rationale) for the feature or bug fix. Unit and/or integration tests are expected for any behaviour change and system tests should be considered for larger changes.* ### Committer Checklist (excluded from commit message) - [ ] Verify design and implementation - [ ] Verify test coverage and CI build status - [ ] Verify documentation (including upgrade notes) ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
4. rhauch commented on pull request #5705: KAFKA-6605 fix NPE in Flatten when optional Struct is null URL: https://github.com/apache/kafka/pull/5705 ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
5. rhauch commented on pull request #5706: KAFKA-6605 fix NPE in Flatten when optional Struct is null - backport… URL: https://github.com/apache/kafka/pull/5706 ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org