

Subject: Process variables are deleted when concurrent execution appears

Hello. Our team tried to upgrade from camunda 7.14.0 to 7.16.0. After that our **tests** (unit and integration) **started to fail**, but when manually checking camunda behavior - it is ok. First version after 7.14.0 where tests started to fail is 7.15-alpha2.

Environment

- Spring Boot 2.6.4
- Java 17
- dependencies
 - "org.camunda.bpm.springboot:camunda-bpm-spring-boot-starter:7.16.0"
 - "org.camunda.bpm.springboot:camunda-bpm-spring-boot-starter-webapp:7.16.0"
 - "org.camunda.bpm.springboot:camunda-bpm-spring-boot-starter-rest:7.16.0"
 - "com.sun.xml.bind:jaxb-impl:3.0.2"
 - "org.graalvm.js:js:22.0.0.2"
 - "org.graalvm.js:js-scriptengine:22.0.0.2"
 - "org.camunda.bpm.assert:camunda-bpm-assert:14.0.0"

Unit Test

Configuration

```
package camunda.shared.configuration;

import java.io.IOException;
import java.util.List;
import javax.sql.DataSource;
import org.camunda.bpm.engine.ProcessEngineConfiguration;
import org.camunda.bpm.engine.impl.cfg.CompositeProcessEnginePlugin;
import org.camunda.bpm.engine.impl.cfg.ProcessEngineConfigurationImpl;
import org.camunda.bpm.engine.impl.cfg.ProcessEnginePlugin;
import org.camunda.bpm.engine.impl.cfg.SpringBeanFactoryProxyMap;
import org.camunda.bpm.engine.impl.el.ExpressionManager;
```

```

import org.camunda.bpm.engine.spring.SpringProcessEngineConfiguration;
import
org.camunda.bpm.extension.process_test_coverage.spring.SpringProcessWithCov
erageEngineConfiguration;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.transaction.PlatformTransactionManager;

@Configuration
public class UnitTestsProcessEngineConfiguration {
    @Bean
    public ProcessEngineConfigurationImpl processEngineConfiguration(
        final List<ProcessEnginePlugin> processEnginePlugins,
        final ExpressionManager expressionManager,
        final PlatformTransactionManager transactionManager,
        final DataSource dataSource,
        final ApplicationContext applicationContext
    ) throws IOException {
        SpringProcessEngineConfiguration config =
            new SpringProcessWithCoverageEngineConfiguration();
        config.setExpressionManager(expressionManager);
        config.setTransactionManager(transactionManager);
        config.setDataSource(dataSource);
        config.setDatabaseSchemaUpdate("true");
        config.setHistory(ProcessEngineConfiguration.HISTORY_FULL);
        config.setJobExecutorActivate(false); // <---
        config.setProcessEnginePlugins(
            processEnginePlugins
        );

        config.getProcessEnginePlugins().add(
            new CompositeProcessEnginePlugin(processEnginePlugins)
        );
        config.setBeans(new SpringBeanFactoryProxyMap(applicationContext));
        return config;
    }
}

```

Here is sample process and test for it:

DAV-2_POC_1.bpmn

```
@Test
@DeployProcess(["DAV-2_POC_1.bpmn"])
fun `POC 1 - variables are deleted`() {
    val builder: ProcessInstantiationBuilder =
this.processEngine.runtimeService
        .createProcessInstanceByKey("DAV-2_POC_1")
    builder.setVariable("featureIssueId", 178825)
    builder.setVariable("implementationCategory", "category-value")
    builder.startBeforeActivity("Implement_feature")
    val processInstance: ProcessInstance = builder.execute();

    // waiting for task to be resolved

BpmnAwareTests.assertThat(processInstance).isWaitingAt("Implement_feature")

    val processExecution: Execution = this.processEngine.runtimeService
        .createExecutionQuery()
        .processInstanceId(processInstance.id)
        .active()
        .list()
        .firstOrNull()!!

    // process variables are there
    assertEquals(178825, getVariable(processExecution.id,
"featureIssueId"))
    assertEquals("category-value", getVariable(processExecution.id,
"implementationCategory"))
    assertNull(getVariable(processExecution.id, "implementationIssue"))

    // triggering script task that will set `issueId`
this.processEngine

.runtimeService.createProcessInstanceModification(processInstance.id)
        .startBeforeActivity("Set_issue_id")
        .execute()

    // `implementationIssue` is set by event
`Implement_feature_issue_id_set` after `issueId` was set
    assertEquals(777, getVariable(processExecution.id,
"implementationIssue"))
}
```

```

        // !!! process variables are not there after event is executed !!!
        assertEquals(178825, getVariable(processExecution.id,
"featureIssueId")) // assert fails
        assertEquals("category-value", getVariable(processExecution.id,
"implementationCategory")) // assert fails
    }

    fun getVariable(executionId: String, variableName: String): Any? {
        return this.processEngine.runtimeService.getVariable(executionId,
variableName)
    }

```

Output (last asserts):

```

Expected :178825
Actual   :null

```

Integration Test

Configuration

```

package camunda.shared.configuration;

import java.io.IOException;
import java.util.List;
import javax.sql.DataSource;
import org.camunda.bpm.engine.ProcessEngineConfiguration;
import org.camunda.bpm.engine.impl.cfg.CompositeProcessEnginePlugin;
import org.camunda.bpm.engine.impl.cfg.ProcessEngineConfigurationImpl;
import org.camunda.bpm.engine.impl.cfg.ProcessEnginePlugin;
import org.camunda.bpm.engine.impl.cfg.SpringBeanFactoryProxyMap;
import org.camunda.bpm.engine.impl.el.ExpressionManager;
import org.camunda.bpm.engine.spring.SpringProcessEngineConfiguration;
import
org.camunda.bpm.extension.process_test_coverage.spring.SpringProcessWithCov
erageEngineConfiguration;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.transaction.PlatformTransactionManager;

@Configuration

```

```

public class IntegrationTestsProcessEngineConfiguration {
    @Bean
    public ProcessEngineConfigurationImpl processEngineConfiguration(
        final List<ProcessEnginePlugin> processEnginePlugins,
        final ExpressionManager expressionManager,
        final PlatformTransactionManager transactionManager,
        final DataSource dataSource,
        final ApplicationContext applicationContext
    ) throws IOException {
        SpringProcessEngineConfiguration config =
            new SpringProcessWithCoverageEngineConfiguration();
        config.setExpressionManager(expressionManager);
        config.setTransactionManager(transactionManager);
        config.setDataSource(dataSource);
        config.setDatabaseSchemaUpdate("true");
        config.setHistory(ProcessEngineConfiguration.HISTORY_FULL);
        config.setJobExecutorActivate(true); // <---
        config.getProcessEnginePlugins().add(
            new CompositeProcessEnginePlugin(processEnginePlugins)
        );
        config.setBeans(new SpringBeanFactoryProxyMap(applicationContext));
        return config;
    }
}

```

Test is the same as above.

Output is the same, variables are removed.

Observations

The place variables are removed

ExecutionEntity.java

```

...
protected void moveVariableTo(VariableInstanceEntity variable,
ExecutionEntity other) {
    if (other.variableStore.containsKey(variable.getName())) {
        // this is executed when ExecutionEntity.replace(PvmExecutionImpl
execution) where comment says: // on compaction, move all variables
        CoreVariableInstance existingInstance =

```

```

other.variableStore.getVariable(variable.getName());
existingInstance.setValue(variable.getTypedValue(false));
invokeVariableLifecycleListenersUpdate(existingInstance, this);
invokeVariableLifecycleListenersDelete( // <-- here
VariableInstanceEntityPersistenceListener.delete()
    variable,
    this,
    Collections.singletonList(getVariablePersistenceListener()));
}
else {
    // this is executed when new concurrent execution is created
    // scenario 2 in PvmExecutionImpl.createConcurrentExecution()
    other.variableStore.addVariable(variable);
}
}
...

```

It seems like it is happening when there are **concurrent execution** takes place.

Manual check

The screenshot shows the Camunda Cockpit interface for a process instance. On the left, the 'Information' tab is active, displaying details for Instance ID: 19c8b012-ab87-11ec-bb4b-4ed577... and Definition Name: DAV-2_POC_1. The main area shows a BPMN diagram for 'Start feature development process' with a 'User task' and a 'Set issue ID' task. Below the diagram, the 'Variables' tab is selected, showing a table of variables.

Name	Type	Value	Scope	Actions
featureIssueId	Long	178825	DAV-2_POC_1	[Red flag] [Trash]
processInstanceAuthorEmail_19c8b012-ab87-...	Null		DAV-2_POC_1	[Red flag] [Trash]
processInstanceTaskId_19c8b012-ab87-11ec-...	Integer	579247	DAV-2_POC_1	[Red flag] [Trash]
processStartDate	Date	2022-03-24T18:28:48	DAV-2_POC_1	[Red flag] [Trash]
processStarterId	Null		DAV-2_POC_1	[Red flag] [Trash]

After 10 seconds

Camunda Cockpit Processes Decisions Human Tasks More

Kermit Root

Dashboard » Processes » DAV-2_POC_1 : 19c8b012-ab87-11ec-bb4b-4ed5776c9c53 : Runtime

Information Filter

Instance ID:
19c8b012-ab87-11ec-bb4b-4ed577...

Business Key:
null

Definition Version:
2

Definition ID:
DAV-2_POC_1:2:0a2e3df0-ab87-11e...

Definition Key:
DAV-2_POC_1

Definition Name:
DAV-2_POC_1

Tenant ID:
null

Deployment ID:
0a2d056e-ab87-11ec-bb4b-4ed577...

Super Process Instance ID:
null

```

graph TD
    Start((Start feature development process)) --> UserTask[User task]
    UserTask --> EndSaving((End saving implementations sue))
    EndSaving --> SetIssueID[Set 'issue ID']
    SetIssueID --> ClockMessage((Clock message))
    
```

Variables Incidents Called Process Instances User Tasks Jobs External Tasks

Name	Type	Value	Scope	Actions
featureIssueId	Long	178825	DAV-2_POC_1	🗑️
implementationIssue	Integer	777	DAV-2_POC_1	🗑️
issueId	Integer	777	DAV-2_POC_1	🗑️
processInstanceAuthorEmail_19c8b012-ab87-...	Null		DAV-2_POC_1	🗑️
processInstanceTaskId_19c8b012-ab87-11ec-...	Integer	579247	DAV-2_POC_1	🗑️
processStartDate	Date	2022-03-24T18:28:48	DAV-2_POC_1	🗑️
processStarterId	Null		DAV-2_POC_1	🗑️

Date and Time displayed in local timezone: Europe/Moscow

Powered by Camunda Platform / v7.16.0

Modified verison of process with same result

DAV-2 POC-1

Start feature development process



'issue ID' is set



End setting implementations



Clock message

