



Fortify Standalone Report Generator

---

# Developer Workbook

---

akka-persistence-testkit



# Table of Contents

- [Executive Summary](#)
- [Project Description](#)
- [Issue Breakdown by Fortify Categories](#)
- [Results Outline](#)

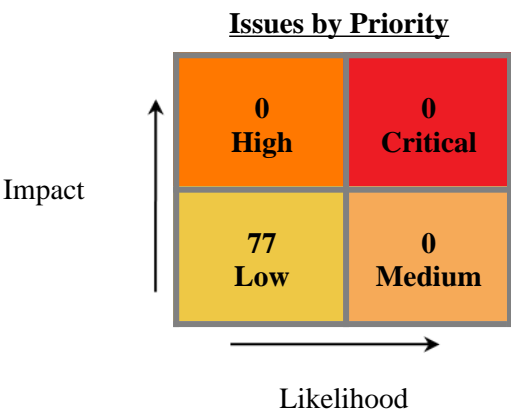


# Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the akka-persistence-testkit project audit. The information contained in this workbook is targeted at project managers and developers.

This section provides an overview of the issues uncovered during analysis.

Project Name:	akka-persistence-testkit
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present



## Top Ten Critical Categories

This project does not contain any critical issues



## Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

### SCA

<b>Date of Last Analysis:</b>	Jun 16, 2022, 11:37 AM	<b>Engine Version:</b>	21.1.1.0009
<b>Host Name:</b>	Jacks-Work-MBP.local	<b>Certification:</b>	VALID
<b>Number of Files:</b>	29	<b>Lines of Code:</b>	1,409

<b>Rulepack Name</b>	<b>Rulepack Version</b>
Fortify Secure Coding Rules, Extended, Java	2022.1.0.0007
Fortify Secure Coding Rules, Core, Scala	2022.1.0.0007
Fortify Secure Coding Rules, Extended, JSP	2022.1.0.0007
Fortify Secure Coding Rules, Core, Android	2022.1.0.0007
Fortify Secure Coding Rules, Extended, Content	2022.1.0.0007
Fortify Secure Coding Rules, Extended, Configuration	2022.1.0.0007
Fortify Secure Coding Rules, Core, Annotations	2022.1.0.0007
Fortify Secure Coding Rules, Community, Cloud	2022.1.0.0007
Fortify Secure Coding Rules, Core, Universal	2022.1.0.0007
Fortify Secure Coding Rules, Core, Java	2022.1.0.0007
Fortify Secure Coding Rules, Community, Universal	2022.1.0.0007



## Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Code Correctness: Constructor Invokes Overridable Function	0	0	0	0 / 25	0 / 25
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 4	0 / 4
Code Correctness: Non-Synchronized Method Overrides Synchronized Method	0	0	0	0 / 48	0 / 48



# Results Outline

## Code Correctness: Constructor Invokes Overridable Function (25 issues)

### Abstract

A constructor of the class calls a function that can be overridden.

### Explanation

When a constructor calls an overridable function, it may allow an attacker to access the `this` reference prior to the object being fully initialized, which can in turn lead to a vulnerability. **Example 1:** The following calls a method that can be overridden.

```
...
class User {
    private String username;
    private boolean valid;
    public User(String username, String password){
        this.username = username;
        this.valid = validateUser(username, password);
    }
    public boolean validateUser(String username, String password){
        //validate user is real and can authenticate
        ...
    }
    public final boolean isValid(){
        return valid;
    }
}
```

Since the function `validateUser` and the class are not `final`, it means that they can be overridden, and then initializing a variable to the subclass that overrides this function would allow bypassing of the `validateUser` functionality. For example:

```
...
class Attacker extends User{
    public Attacker(String username, String password){
        super(username, password);
    }
    public boolean validateUser(String username, String password){
        return true;
    }
}
...
class MainClass{
    public static void main(String[] args){
        User hacker = new Attacker("Evil", "Hacker");
        if (hacker.isValid()){
            System.out.println("Attack successful!");
        }else{
            System.out.println("Attack failed");
        }
    }
}
```

The code in Example 1 prints "Attack successful!", since the `Attacker` class overrides the `validateUser()` function that is called from the constructor of the superclass `User`, and Java will first look in the subclass for functions called from the constructor.



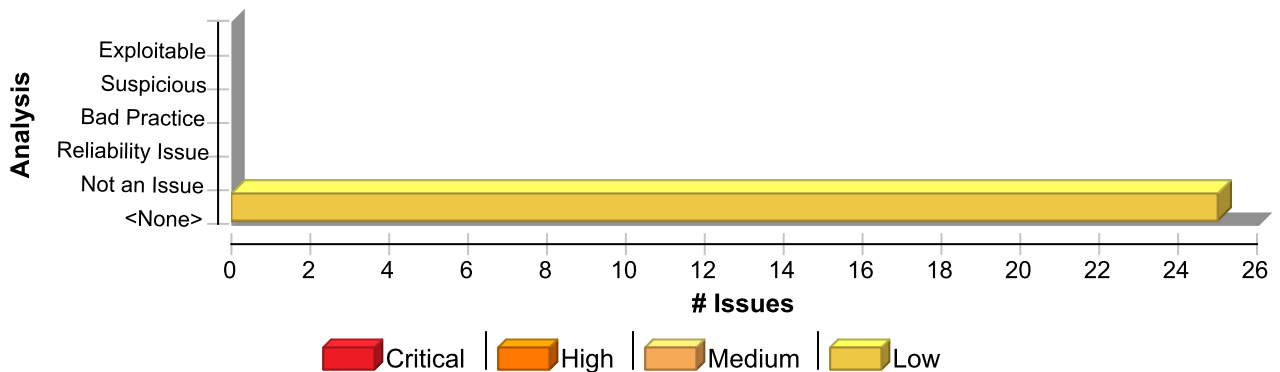
### Recommendation

Constructors should not call functions that can be overridden, either by specifying them as `final`, or specifying the class as `final`. Alternatively if this code is only ever needed in the constructor, the `private` access specifier can be used, or the logic could be placed directly into the constructor of the superclass. **Example 2:** The following makes the class `final` to prevent the function from being overridden elsewhere.

```
...
final class User {
    private String username;
    private boolean valid;
    public User(String username, String password){
        this.username = username;
        this.valid = validateUser(username, password);
    }
    private boolean validateUser(String username, String password){
        //validate user is real and can authenticate
        ...
    }
    public final boolean isValid(){
        return valid;
    }
}
```

This example specifies the class as `final`, so that it cannot be subclassed, and changes the `validateUser()` function to `private`, since it is not needed elsewhere in this application. This is programming defensively, since at a later date it may be decided that the `User` class needs to be subclassed, which would result in this vulnerability reappearing if the `validateUser()` function was not set to `private`.

### Issue Summary



### Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Constructor Invokes Overridable Function	25	0	0	25
<b>Total</b>	<b>25</b>	<b>0</b>	<b>0</b>	<b>25</b>

<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
Package: akka.persistence.testkit	
<b>PersistenceTestKitPlugin.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
Issue Details	



<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit	
<b>PersistenceTestKitPlugin.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: PluginId  
**Enclosing Method:** PersistenceTestKitPlugin()  
**File:** PersistenceTestKitPlugin.scala:91  
**Taint Flags:**

```

88 def getInstance() = this
89
90 val config: Config = ConfigFactory.parseMap(
91 Map(
92 "akka.persistence.journal.plugin" -> PluginId,
93 s"$PluginId.class" -> s"${classOf[PersistenceTestKitPlugin].getName}").asJava)
94

```

<b>PersistenceTestKitPlugin.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: PluginId  
**Enclosing Method:** PersistenceTestKitPlugin()  
**File:** PersistenceTestKitPlugin.scala:91  
**Taint Flags:**

```

88 def getInstance() = this
89
90 val config: Config = ConfigFactory.parseMap(
91 Map(
92 "akka.persistence.journal.plugin" -> PluginId,
93 s"$PluginId.class" -> s"${classOf[PersistenceTestKitPlugin].getName}").asJava)
94

```

<b>PersistenceTestKitPlugin.scala, line 148 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)





**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit**PersistenceTestKitPlugin.scala, line 148 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details**

**Sink:** FunctionCall: PluginId  
**Enclosing Method:** PersistenceTestKitDurableStateStorePlugin()  
**File:** PersistenceTestKitPlugin.scala:148  
**Taint Flags:**

```
145  
146 def getInstance() = this  
147  
148 val config: Config = ConfigFactory.parseMap(Map("akka.persistence.state.plugin" -> PluginId).asJava)  
149 }  
150  
151 undefined
```

**PersistenceTestKitPlugin.scala, line 132 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** FunctionCall: PluginId  
**Enclosing Method:** PersistenceTestKitSnapshotPlugin()  
**File:** PersistenceTestKitPlugin.scala:132  
**Taint Flags:**

```
129 def getInstance() = this  
130  
131 val config: Config = ConfigFactory.parseMap(  
132 Map(  
133 "akka.persistence.snapshot-store.plugin" -> PluginId,  
134 s"$PluginId.class" -> classOf[PersistenceTestKitSnapshotPlugin].getName,  
135 s"$PluginId.snapshot-is-optional" -> false // fallback isn't used by the testkit
```

**PersistenceTestKitPlugin.scala, line 132 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** FunctionCall: PluginId  
**Enclosing Method:** PersistenceTestKitSnapshotPlugin()



**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit**PersistenceTestKitPlugin.scala, line 132 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** PersistenceTestKitPlugin.scala:132**Taint Flags:**

```
129 def getInstance() = this
130
131 val config: Config = ConfigFactory.parseMap(
132 Map(
133 "akka.persistence.snapshot-store.plugin" -> PluginId,
134 s"$PluginId.class" -> classOf[PersistenceTestKitSnapshotPlugin].getName,
135 s"$PluginId.snapshot-is-optional" -> false // fallback isn't used by the testkit
```

**Package:** akka.persistence.testkit.internal**internal/EventSourcedBehaviorTestKitImpl.scala, line 106 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: snapshotTestKit**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:106**Taint Flags:**

```
103 if (system.settings.config.getString("akka.persistence.snapshot-store.plugin") != "")
104 Some(SnapshotTestKit(system))
105 else None
106 snapshotTestKit.foreach(_clearAll())
107
108 private val queries =
109 PersistenceQuery(system).readJournalFor[CurrentEventsByPersistenceIdQuery](PersistenceTestKitReadJournal.Identifier)
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 116 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: probe**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:116**Taint Flags:**

**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit.internal**internal/EventSourcedBehaviorTestKitImpl.scala, line 116 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
113 private var actor: ActorRef[Command] = actorTestKit.spawn(behavior)
114 private def internalActor = actor.unsafeUpcast[Any]
115 private val persistenceId: PersistenceId = {
116   internalActor ! EventSourcedBehaviorImpl.GetPersistenceId(probe.ref)
117 } try {
118   probe.expectMessageType[PersistenceId]
119 } catch {
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: persistenceTestKit**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:100**Taint Flags:**

```
97 }
98
99 override val persistenceTestKit: PersistenceTestKit = PersistenceTestKit(system)
100 persistenceTestKit.clearAll()
101
102 override val snapshotTestKit: Option[SnapshotTestKit] =
103 if (system.settings.config.getString("akka.persistence.snapshot-store.plugin") != "")
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 115 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: liftedTree1**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:115**Taint Flags:**

```
112 private val stateProbe = actorTestKit.createTestProbe[GetStateReply[State]]()
113 private var actor: ActorRef[Command] = actorTestKit.spawn(behavior)
114 private def internalActor = actor.unsafeUpcast[Any]
```



<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 115 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

```

115 private val persistenceId: PersistenceId = {
116   internalActor ! EventSourcedBehaviorImpl.GetPersistenceId(probe.ref)
117   try {
118     probe.expectMessageType[PersistenceId]

```

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
<b>Issue Details</b>	

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: system  
**Enclosing Method:** EventSourcedBehaviorTestKitImpl()  
**File:** internal/EventSourcedBehaviorTestKitImpl.scala:91  
**Taint Flags:**

```

88 import EventSourcedBehaviorTestKitImpl._
89
90 private def system: ActorSystem[_] = actorTestKit.system
91 if (system.settings.config.getBoolean("akka.persistence.testkit.events.serialize") ||
92   system.settings.config.getBoolean("akka.persistence.testkit.snapshots.serialize")) {
93   system.log.warn(
94     "Persistence TestKit serialization enabled when using EventSourcedBehaviorTestKit, this is not intended. " +

```

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 92 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
<b>Issue Details</b>	

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: system  
**Enclosing Method:** EventSourcedBehaviorTestKitImpl()  
**File:** internal/EventSourcedBehaviorTestKitImpl.scala:92  
**Taint Flags:**

```

89
90 private def system: ActorSystem[_] = actorTestKit.system
91 if (system.settings.config.getBoolean("akka.persistence.testkit.events.serialize") ||
92   system.settings.config.getBoolean("akka.persistence.testkit.snapshots.serialize")) {
93   system.log.warn(
94     "Persistence TestKit serialization enabled when using EventSourcedBehaviorTestKit, this is not intended. " +

```



<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 92 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

95 "make sure you create the system used in the test with the config from EventSourcedBehaviorTestKit.config " +

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: system

**Enclosing Method:** EventSourcedBehaviorTestKitImpl()

**File:** internal/EventSourcedBehaviorTestKitImpl.scala:93

**Taint Flags:**

```

90 private def system: ActorSystem[_] = actorTestKit.system
91 if (system.settings.config.getBoolean("akka.persistence.testkit.events.serialize") ||
92 system.settings.config.getBoolean("akka.persistence.testkit.snapshots.serialize")) {
93 system.log.warn(
94 "Persistence TestKit serialization enabled when using EventSourcedBehaviorTestKit, this is not intended. " +
95 "make sure you create the system used in the test with the config from EventSourcedBehaviorTestKit.config " +
96 "as described in the docs https://doc.akka.io/docs/akka/current/typed/persistence-testing.html#unit-testing")

```

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 99 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: system

**Enclosing Method:** EventSourcedBehaviorTestKitImpl()

**File:** internal/EventSourcedBehaviorTestKitImpl.scala:99

**Taint Flags:**

```

96 "as described in the docs https://doc.akka.io/docs/akka/current/typed/persistence-testing.html#unit-testing")
97 }
98
99 override val persistenceTestKit: PersistenceTestKit = PersistenceTestKit(system)
100 persistenceTestKit.clearAll()
101
102 override val snapshotTestKit: Option[SnapshotTestKit] =

```



**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit.internal**internal/EventSourcedBehaviorTestKitImpl.scala, line 103 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: system**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:103**Taint Flags:**

```
100 persistenceTestKit.clearAll()
101
102 override val snapshotTestKit: Option[SnapshotTestKit] =
103 if (system.settings.config.getString("akka.persistence.snapshot-store.plugin") != "")
104 Some(SnapshotTestKit(system))
105 else None
106 snapshotTestKit.foreach(_._clearAll())
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 104 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: system**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:104**Taint Flags:**

```
101
102 override val snapshotTestKit: Option[SnapshotTestKit] =
103 if (system.settings.config.getString("akka.persistence.snapshot-store.plugin") != "")
104 Some(SnapshotTestKit(system))
105 else None
106 snapshotTestKit.foreach(_._clearAll())
107
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 108 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit.internal**internal/EventSourcedBehaviorTestKitImpl.scala, line 108 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: system**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:108**Taint Flags:**

```
105 else None
106 snapshotTestKit.foreach(_clearAll())
107
108 private val queries =
109 PersistenceQuery(system).readJournalFor[CurrentEventsByPersistenceIdQuery](PersistenceTestKitReadJournal.Identifier)
110
111 private val probe = actorTestKit.createTestProbe[Any]()
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 124 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: system**Enclosing Method:** EventSourcedBehaviorTestKitImpl()**File:** internal/EventSourcedBehaviorTestKitImpl.scala:124**Taint Flags:**

```
121 throw new IllegalArgumentException("Only EventSourcedBehavior, or nested EventSourcedBehavior allowed.")
122 }
123 }
124 private val serializationTestKit = new SerializationTestKit(system)
125
126 private var emptyStateVerified = false
127
```

**internal/EventSourcedBehaviorTestKitImpl.scala, line 116 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: internalActor**Enclosing Method:** EventSourcedBehaviorTestKitImpl()

<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
<b>Package: akka.persistence.testkit.internal</b>	
<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 116 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

**File:** internal/EventSourcedBehaviorTestKitImpl.scala:116

**Taint Flags:**

```

113 private var actor: ActorRef[Command] = actorTestKit.spawn(behavior)
114 private def internalActor = actor.unsafeUpcast[Any]
115 private val persistenceId: PersistenceId = {
116 internalActor ! EventSourcedBehaviorImpl.GetPersistenceId(probe.ref)
117 try {
118 probe.expectMessageType[PersistenceId]
119 } catch {

```

<b>Package: akka.persistence.testkit.query.scaladsl</b>	
<b>query/scaladsl/PersistenceTestKitReadJournal.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: log

**Enclosing Method:** PersistenceTestKitReadJournal()

**File:** query/scaladsl/PersistenceTestKitReadJournal.scala:50

**Taint Flags:**

```

47 private val storage: EventStorage = {
48 // use shared path up to before `query` to identify which inmem journal we are addressing
49 val storagePluginId = configPath.replaceAll("query$", "journal")
50 log.debug("Using in memory storage [{}] for test kit read journal", storagePluginId)
51 InMemStorageExtension(system).storageFor(storagePluginId)
52 }
53

```

<b>Package: akka.persistence.testkit.scaladsl</b>	
<b>scaladsl/PersistenceTestKit.scala, line 454 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: settings

**Enclosing Method:** PersistenceTestKit()





<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.scaladsl	
<b>scaladsl/PersistenceTestKit.scala, line 454 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>

**File:** scaladsl/PersistenceTestKit.scala:454

**Taint Flags:**

```

451
452 override private[testkit] val Policies = EventStorage.JournalPolicies
453
454 override private[testkit] val pollInterval: FiniteDuration = settings.pollInterval
455
456 override private[testkit] val maxTimeout: FiniteDuration = settings.assertTimeout
457

```

<b>scaladsl/PersistenceTestKit.scala, line 456 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: settings

**Enclosing Method:** PersistenceTestKit()

**File:** scaladsl/PersistenceTestKit.scala:456

**Taint Flags:**

```

453
454 override private[testkit] val pollInterval: FiniteDuration = settings.pollInterval
455
456 override private[testkit] val maxTimeout: FiniteDuration = settings.assertTimeout
457
458 override def rejectNextNPersisted(persistenceId: String, n: Int, cause: Throwable): Unit =
459 rejectNextNOpsCond((pid, op) => pid == persistenceId && op.isInstanceOf[WriteEvents], n, cause)

```

<b>scaladsl/PersistenceTestKit.scala, line 345 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: settings

**Enclosing Method:** SnapshotTestKit()

**File:** scaladsl/PersistenceTestKit.scala:345

**Taint Flags:**

```

342

```



**Code Correctness: Constructor Invokes Overridable Function****Low****Package:** akka.persistence.testkit.scaladsl**scaladsl/PersistenceTestKit.scala, line 345 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
343 override private[testkit] val probe = TestProbe()(system)
344
345 override private[testkit] val pollInterval: FiniteDuration = settings.pollInterval
346
347 override private[testkit] val maxTimeout: FiniteDuration = settings.assertTimeout
348
```

**scaladsl/PersistenceTestKit.scala, line 347 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** FunctionCall: settings  
**Enclosing Method:** SnapshotTestKit()  
**File:** scaladsl/PersistenceTestKit.scala:347  
**Taint Flags:**

```
344
345 override private[testkit] val pollInterval: FiniteDuration = settings.pollInterval
346
347 override private[testkit] val maxTimeout: FiniteDuration = settings.assertTimeout
348
349 override private[testkit] val Policies = SnapshotStorage.SnapshotPolicies
350
```

**Package:** akka.persistence.testkit.state.scaladsl**state/scaladsl/PersistenceTestKitDurableStateStore.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** FunctionCall: EarliestOffset  
**Enclosing Method:** PersistenceTestKitDurableStateStore()  
**File:** state/scaladsl/PersistenceTestKitDurableStateStore.scala:53  
**Taint Flags:**

```
50 .run()
51
52 private val EarliestOffset = 0L
```



<b>Code Correctness: Constructor Invokes Overridable Function</b>	<b>Low</b>
---	------------

Package: akka.persistence.testkit.state.scaladsl

<b>state/scaladsl/PersistenceTestKitDurableStateStore.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

```
53 private val lastGlobalOffset = new AtomicLong(EarliestOffset)
```

```
54
```

```
55 override def getObject(persistenceId: String): Future[GetObjectResult[A]] = this.synchronized {
```

```
56 Future.successful(store.get(persistenceId) match {
```

<b>state/scaladsl/PersistenceTestKitDurableStateStore.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** FunctionCall: sys

**Enclosing Method:** PersistenceTestKitDurableStateStore()

**File:** state/scaladsl/PersistenceTestKitDurableStateStore.scala:50

**Taint Flags:**

```
47 ActorSource
```

```
48 .actorRef[Record[A]](PartialFunction.empty, PartialFunction.empty, 256, OverflowStrategy.dropHead)
```

```
49 .toMat(BroadcastHub.sink)(Keep.both)
```

```
50 .run()
```

```
51
```

```
52 private val EarliestOffset = 0L
```

```
53 private val lastGlobalOffset = new AtomicLong(EarliestOffset)
```



## Code Correctness: Non-Static Inner Class Implements Serializable (4 issues)

### Abstract

Inner classes implementing `java.io.Serializable` may cause problems and leak information from the outer class.

### Explanation

Serialization of inner classes lead to serialization of the outer class, therefore possibly leaking information or leading to a runtime error if the outer class is not serializable. As well as this, serializing inner classes may cause platform dependencies since the Java compiler creates synthetic fields in order to implement inner classes, but these are implementation dependent, and may vary from compiler to compiler. **Example 1:** The following code allows serialization of an inner class.

```
...
class User implements Serializable {
    private int accessLevel;
    class Registrator implements Serializable {
        ...
    }
}
```

In Example 1, when the inner class `Registrator` is serialized, it will also serialize the field `accessLevel` from the outer class `User`.

### Recommendation

When using inner classes, they should not be serialized, or they should be changed to static-nested classes, since these do not have the drawbacks that non-static inner classes have when serialized. When a nested class is static it inherently has no association with instance variables (including those of the outer class), and would not cause serialization of the outer class. **Example 2:** The following code changes the example in Example 1, by stopping the inner class from implementing `java.io.Serializable`.

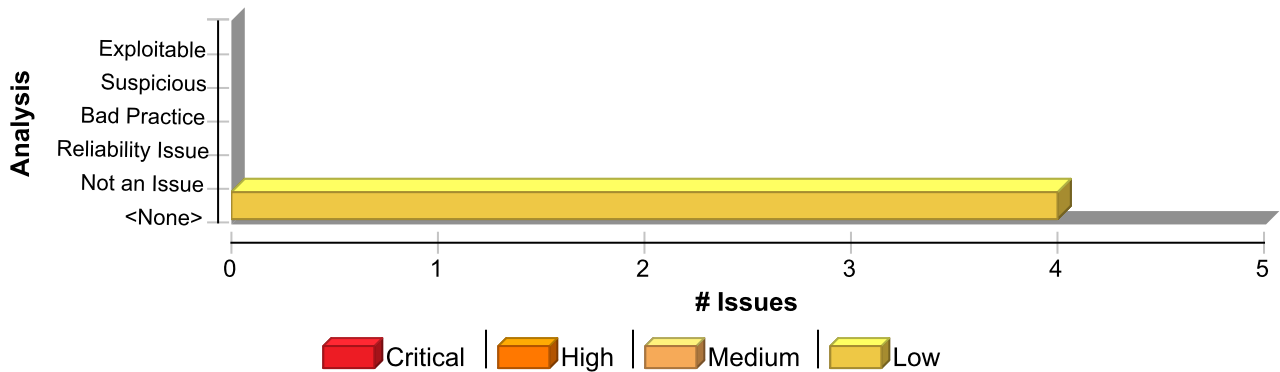
```
...
class User implements Serializable {
    private int accessLevel;
    class Registrator {
        ...
    }
}
```

**Example 2:** The following code changes the example in Example 1, by making the inner class into a static-nested class.

```
...
class User implements Serializable {
    private int accessLevel;
    static class Registrator implements Serializable {
        ...
    }
}
```

### Issue Summary





## Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Non-Static Inner Class Implements Serializable	4	0	0	4
<b>Total</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>4</b>

<b>Code Correctness: Non-Static Inner Class Implements Serializable</b>	<b>Low</b>
---	------------

Package: akka.persistence.testkit

<b>PersistenceTestKitPlugin.scala, line 95 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>
---	------------

### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

### Sink Details

**Sink:** Class: PersistenceTestKitPlugin\$Write  
**File:** PersistenceTestKitPlugin.scala:95  
**Taint Flags:**

```

92 "akka.persistence.journal.plugin" -> PluginId,
93 s"$PluginId.class" -> s"${classOf[PersistenceTestKitPlugin].getName}").asJava)
94
95 private[testkit] case class Write(persistenceId: String, toSequenceNr: Long)
96
97 }
98

```

Package: akka.persistence.testkit.internal

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 76 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>
---	------------

### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

### Sink Details



<b>Code Correctness: Non-Static Inner Class Implements Serializable</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 76 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>

**Sink:** Class: EventSourcedBehaviorTestKitImpl\$RestartResultImpl  
**File:** internal/EventSourcedBehaviorTestKitImpl.scala:76  
**Taint Flags:**

```

73 override def hasNoReply: Boolean = replyOption.isEmpty
74 }
75
76 final case class RestartResultImpl[State](state: State) extends RestartResult[State]
77 }
78
79 /**

```

<b>internal/EventSourcedBehaviorTestKitImpl.scala, line 37 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Class: EventSourcedBehaviorTestKitImpl\$CommandResultImpl  
**File:** internal/EventSourcedBehaviorTestKitImpl.scala:37  
**Taint Flags:**

```

34 * INTERNAL API
35 */
36 @InternalApi private[akka] object EventSourcedBehaviorTestKitImpl {
37 final case class CommandResultImpl[Command, Event, State, Reply](
38 command: Command,
39 events: immutable.Seq[Event],
40 state: State,

```

<b>internal/SerializedEventStorageImpl.scala, line 17 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Class: SerializedEventStorageImpl\$Serialized  
**File:** internal/SerializedEventStorageImpl.scala:17  
**Taint Flags:**

```

14
15 @InternalApi

```



<b>Code Correctness: Non-Static Inner Class Implements Serializable</b>	<b>Low</b>
<b>Package: akka.persistence.testkit.internal</b>	
<b>internal/SerializedEventStorageImpl.scala, line 17 (Code Correctness: Non-Static Inner Class Implements Serializable)</b>	<b>Low</b>

```

16 private[testkit] object SerializedEventStorageImpl {
17   case class Serialized(
18     persistenceId: String,
19     sequenceNr: Long,
20     payloadSerId: Int,

```

## Code Correctness: Non-Synchronized Method Overrides Synchronized Method (48 issues)

### Abstract

Synchronized methods should not be overridden with non-synchronized methods.

### Explanation

A parent class declared the method `synchronized`, guaranteeing correct behavior when multiple threads access the same instance. All overriding methods should also be declared `synchronized`, otherwise unexpected behavior may occur. **Example 1:** In the following code, the class `Foo` overrides the class `Bar` but does not declare the method `synchronizedMethod` to be `synchronized`:

```
public class Bar {  
    public synchronized void synchronizedMethod() {  
        for (int i=0; i<10; i++) System.out.print(i);  
        System.out.println();  
    }  
}
```

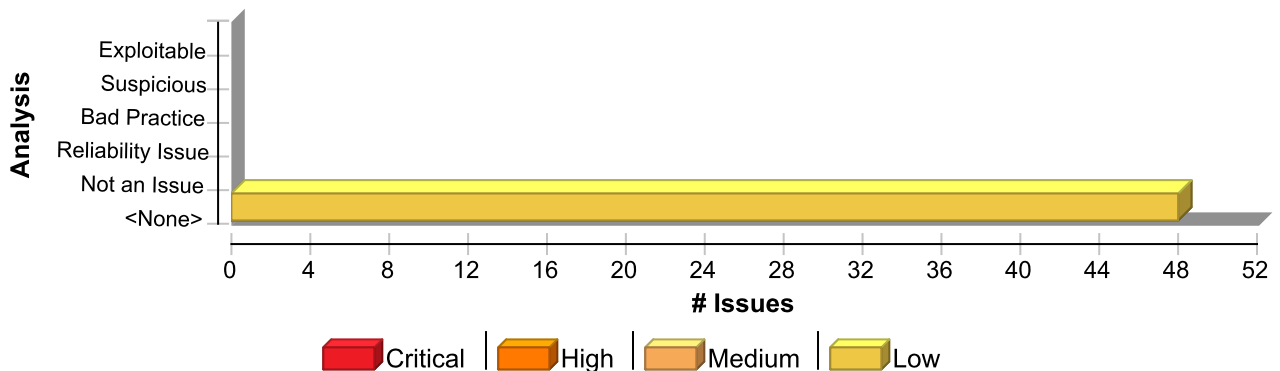
```
public class Foo extends Bar {  
    public void synchronizedMethod() {  
        for (int i=0; i<10; i++) System.out.print(i);  
        System.out.println();  
    }  
}
```

In this case, an instance of `Foo` could be cast to type `Bar`. If the same instance is given to two separate threads and `synchronizedMethod` is executed repeatedly, the behavior will be unpredictable.

### Recommendation

If the parent method is `synchronized`, the method must be declared `synchronized`.

### Issue Summary



### Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Non-Synchronized Method Overrides Synchronized Method	48	0	0	48
<b>Total</b>	<b>48</b>	<b>0</b>	<b>0</b>	<b>48</b>





<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
--	------------

Package: akka.persistence.testkit.internal

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: findOneByIndex  
**Enclosing Method:** findOneByIndex()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: updateOrSetNew  
**Enclosing Method:** updateOrSetNew()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17   override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)



**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Sink Details**

**Sink:** Function: firstInExpectNextQueue  
**Enclosing Method:** firstInExpectNextQueue()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```
14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19   override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20 }
```

**internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** Function: getHighestSeqNumber  
**Enclosing Method:** getHighestSeqNumber()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```
31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)
```

**internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details**

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

**Sink Details**

**Sink:** Function: read  
**Enclosing Method:** read()



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

**File:** internal/SimpleEventStorageImpl.scala:15

**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: getHighestSeqNumber

**Enclosing Method:** getHighestSeqNumber()

**File:** internal/SimpleSnapshotStorageImpl.scala:15

**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17   override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: updateOrSetNew

**Enclosing Method:** updateOrSetNew()

**File:** internal/SerializedEventStorageImpl.scala:34

**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package: akka.persistence.testkit.internal</b>	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

```

32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAllPreservingSeqNumbers  
**Enclosing Method:** clearAllPreservingSeqNumbers()  
**File:** internal/SimpleEventStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAllPreservingSeqNumbers  
**Enclosing Method:** clearAllPreservingSeqNumbers()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

```

35 override type InternalRepr = Serialized
36
37 private lazy val serialization = SerializationExtension(system)

```

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: findOneByIndex  
**Enclosing Method:** findOneByIndex()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```

14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SimpleEventStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17 override type InternalRepr = PersistentRepr
18

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
--	------------

Package: akka.persistence.testkit.internal

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeKey  
**Enclosing Method:** removeKey()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```

14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeKey  
**Enclosing Method:** removeKey()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17 override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

#### Sink Details

**Sink:** Function: readAll  
**Enclosing Method:** readAll()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17   override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAll  
**Enclosing Method:** clearAll()



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

**File:** internal/SerializedEventStorageImpl.scala:34

**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: firstInExpectNextQueue

**Enclosing Method:** firstInExpectNextQueue()

**File:** internal/SerializedEventStorageImpl.scala:34

**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeKey

**Enclosing Method:** removeKey()

**File:** internal/SimpleEventStorageImpl.scala:15

**Taint Flags:**

```

12 * INTERNAL API

```





<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package: akka.persistence.testkit.internal</b>	
<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
<pre> 13 */ 14 @InternalApi 15 private[testkit] class SimpleEventStorageImpl extends EventStorage { 16 17   override type InternalRepr = PersistentRepr 18 </pre>	
<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
<b>Issue Details</b>	
<b>Kingdom:</b> Code Quality <b>Scan Engine:</b> SCA (Structural)	
<b>Sink Details</b>	
<b>Sink:</b> Function: readAll <b>Enclosing Method:</b> readAll() <b>File:</b> internal/SimpleEventStorageImpl.scala:15 <b>Taint Flags:</b>	
<pre> 12 * INTERNAL API 13 */ 14 @InternalApi 15 private[testkit] class SimpleEventStorageImpl extends EventStorage { 16 17   override type InternalRepr = PersistentRepr 18 </pre>	
<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
<b>Issue Details</b>	
<b>Kingdom:</b> Code Quality <b>Scan Engine:</b> SCA (Structural)	
<b>Sink Details</b>	
<b>Sink:</b> Function: removeFirstInExpectNextQueue <b>Enclosing Method:</b> removeFirstInExpectNextQueue() <b>File:</b> internal/SimpleSnapshotStorageImpl.scala:15 <b>Taint Flags:</b>	
<pre> 12 * INTERNAL API 13 */ 14 @InternalApi 15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage { </pre>	



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

```

16
17 override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAll  
**Enclosing Method:** clearAll()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17 override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35 override type InternalRepr = Serialized
36
37 private lazy val serialization = SerializationExtension(system)

```



**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: add**Enclosing Method:** add()**File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

```
14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20
```

**internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: add**Enclosing Method:** add()**File:** internal/SimpleSnapshotStorageImpl.scala:15**Taint Flags:**

```
12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17 override type InternalRepr = (SnapshotMetadata, Any)
18
```

**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Sink Details****Sink:** Function: clearAll**Enclosing Method:** clearAll()**File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

14 \* INTERNAL API

15 \*/

16 @InternalApi

17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {

18

19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])

20

**internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: readAll**Enclosing Method:** readAll()**File:** internal/SimpleSnapshotStorageImpl.scala:15**Taint Flags:**

12 \* INTERNAL API

13 \*/

14 @InternalApi

15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {

16

17 override type InternalRepr = (SnapshotMetadata, Any)

18

**internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: add**Enclosing Method:** add()

<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

**File:** internal/SimpleEventStorageImpl.scala:15

**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeFirstInExpectNextQueue

**Enclosing Method:** removeFirstInExpectNextQueue()

**File:** internal/SerializedEventStorageImpl.scala:34

**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality

**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: findOneByIndex

**Enclosing Method:** findOneByIndex()

**File:** internal/SimpleEventStorageImpl.scala:15

**Taint Flags:**

```

12 * INTERNAL API

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package: akka.persistence.testkit.internal</b>	
<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

```

13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: findOneByIndex  
**Enclosing Method:** findOneByIndex()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17   override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

```

16
17 override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```

14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAll  
**Enclosing Method:** clearAll()  
**File:** internal/SimpleEventStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17 override type InternalRepr = PersistentRepr
18

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
--	------------

Package: akka.persistence.testkit.internal

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: readAll  
**Enclosing Method:** readAll()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```

14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeFirstInExpectNextQueue  
**Enclosing Method:** removeFirstInExpectNextQueue()  
**File:** internal/SimpleEventStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17 override type InternalRepr = PersistentRepr
18

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)





**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Sink Details****Sink:** Function: getHighestSeqNumber**Enclosing Method:** getHighestSeqNumber()**File:** internal/SimpleEventStorageImpl.scala:15**Taint Flags:**

```
12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18 }
```

**internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: firstInExpectNextQueue**Enclosing Method:** firstInExpectNextQueue()**File:** internal/SimpleEventStorageImpl.scala:15**Taint Flags:**

```
12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17   override type InternalRepr = PersistentRepr
18 }
```

**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: read**Enclosing Method:** read()

**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

```
14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19   override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20 }
```

**internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: add**Enclosing Method:** add()**File:** internal/SerializedEventStorageImpl.scala:34**Taint Flags:**

```
31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)
```

**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: getHighestSeqNumber**Enclosing Method:** getHighestSeqNumber()**File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

```
14 * INTERNAL API
```



**Code Correctness: Non-Synchronized Method Overrides Synchronized Method****Low****Package:** akka.persistence.testkit.internal**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low**

```
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19   override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20 }
```

**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: clearAllPreservingSeqNumbers**Enclosing Method:** clearAllPreservingSeqNumbers()**File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

```
14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19   override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20 }
```

**internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Function: updateOrSetNew**Enclosing Method:** updateOrSetNew()**File:** internal/SerializedSnapshotStorageImpl.scala:17**Taint Flags:**

```
14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
--	------------

Package: akka.persistence.testkit.internal

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

```

18
19 override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```

<b>internal/SimpleEventStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: updateOrSetNew  
**Enclosing Method:** updateOrSetNew()  
**File:** internal/SimpleEventStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleEventStorageImpl extends EventStorage {
16
17 override type InternalRepr = PersistentRepr
18

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: clearAllPreservingSeqNumbers  
**Enclosing Method:** clearAllPreservingSeqNumbers()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17 override type InternalRepr = (SnapshotMetadata, Any)
18

```



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeKey  
**Enclosing Method:** removeKey()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SimpleSnapshotStorageImpl.scala, line 15 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: firstInExpectNextQueue  
**Enclosing Method:** firstInExpectNextQueue()  
**File:** internal/SimpleSnapshotStorageImpl.scala:15  
**Taint Flags:**

```

12 * INTERNAL API
13 */
14 @InternalApi
15 private[testkit] class SimpleSnapshotStorageImpl extends SnapshotStorage {
16
17   override type InternalRepr = (SnapshotMetadata, Any)
18

```

<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
---	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)



<b>Code Correctness: Non-Synchronized Method Overrides Synchronized Method</b>	<b>Low</b>
<b>Package:</b> akka.persistence.testkit.internal	
<b>internal/SerializedEventStorageImpl.scala, line 34 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>

#### Sink Details

**Sink:** Function: read  
**Enclosing Method:** read()  
**File:** internal/SerializedEventStorageImpl.scala:34  
**Taint Flags:**

```

31 * FIXME, once we add serializers for metadata serialize the metadata payload if present
32 */
33 @InternalApi
34 private[testkit] class SerializedEventStorageImpl(system: ActorSystem) extends EventStorage {
35   override type InternalRepr = Serialized
36
37   private lazy val serialization = SerializationExtension(system)

```

<b>internal/SerializedSnapshotStorageImpl.scala, line 17 (Code Correctness: Non-Synchronized Method Overrides Synchronized Method)</b>	<b>Low</b>
--	------------

#### Issue Details

**Kingdom:** Code Quality  
**Scan Engine:** SCA (Structural)

#### Sink Details

**Sink:** Function: removeFirstInExpectNextQueue  
**Enclosing Method:** removeFirstInExpectNextQueue()  
**File:** internal/SerializedSnapshotStorageImpl.scala:17  
**Taint Flags:**

```

14 * INTERNAL API
15 */
16 @InternalApi
17 private[testkit] class SerializedSnapshotStorageImpl(system: ActorSystem) extends SnapshotStorage {
18
19   override type InternalRepr = (SnapshotMetadata, String, Int, Array[Byte])
20

```



