

## OOPSLA 23 Artifact

This document provides evaluation guidelines for **Paper #374: Initializing Global Objects: Time and Order**.

The artifact includes an implementation of the global object initialization checking algorithm described in the paper, integrated into Dotty, the Scala 3 compiler. This document helps verify the following:

- The code snippets in the paper are either rejected or accepted by the checker as expected.
- The Scala open issues mentioned in Appendix D are fixed.
- The case study in Section 6 can be reproduced.

## Getting Started

The artifact is a Docker image. The reviewer is invited to install Docker and make sure it is working normally. The following official guide can help with the installation of Docker: <https://docs.docker.com/get-docker/>.

**Memory Requirement:** Normally the default configuration just works. In case there are memory problems, please set the memory quota for the VM to 4GB.

Once Docker is installed on the system:

1. **Download the Docker image:** `docker pull qata/oopsla23-artifact:latest`
2. **Run the Docker image:** `docker run -it qata/oopsla23-artifact:latest`

## Play with Examples

Test cases can be found in the directory `/home/dotty/tests/init-global`:

- Test cases in `tests/init-global/pos` are expected to pass the check with no warnings.
- Test cases in `tests/init-global/neg` are expected to be rejected by the checker with warnings.

We can run the global initialization checker on a test file as follows:

```
cd /home/dotty
bin/scalac -Ysafe-init-global -d /home/tmp tests/init-global/neg/mutable-read7.scala
```

The compiler is expected to produce the following warning:

```
-- Warning: tests/init-global/neg/mutable-read7.scala:7:17
-----
7 |   if (Positioned.debug) { // error
  |   ^^^^^^^^^^^^^^^^^^^^^^
  | Reading mutable state of object Positioned during initialization of
  | object Trees.
  | Reading mutable state of other static objects is forbidden as it breaks
  | initialization-time irrelevance. Calling trace:
  | -> object Trees:      [ mutable-read7.scala:11 ]
  |   ^
  | -> val emptyTree = new Tree [ mutable-read7.scala:13 ]
  |                   ^^^^^^^^^
  | -> class Tree extends Positioned [ mutable-read7.scala:12 ]
  |   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  | -> abstract class Positioned: [ mutable-read7.scala:6 ]
  |   ^
  | -> if (Positioned.debug) { // error [ mutable-read7.scala:7 ]
  |   ^^^^^^^^^^^^^^^^^^^^^^
1 | warning found
```

The reviewer is invited to play with other examples in the `/home/dotty/tests/init-global` directory.

## Verify Code Snippets in the Paper

All code snippets from the paper are located in `/home/snippets`. The file name of a snippet has the form `2.3-1-neg.scala`, which can be read as follows:

- `2.3` means that the snippet comes from **Section 2.3** of the paper.
- `1` tells that it is the **1st** snippet in the corresponding section.
- `neg` indicates that the snippet is expected to be **rejected** by the checker with warnings.

The suffix `pos` indicates that the snippet is expected to **pass** the check with no warnings.

We can check a code snippet as follows:

```
cd /home/dotty
bin/scalac -Ysafe-init-global -d /home/tmp /home/snippets/2.3-1-neg.scala
```

We can check all snippets with the following command:

```
/home/test-all.sh
```

The script will check each file and finally print the results to console as a table. The meanings of each column are as follows:

- **Snippet Name:** The name of the snippet.

- **Expected:** Will be `warning` if the test should produce initialization warnings, and `no warning` if the test should pass the check.
- **Actual:** Will be `warning` if the checker produced warnings on this test, and `no warning` if the it did not.
- **Status:** Will be `pass` if the Expected column matches the Actual column, and `fail` otherwise.

The Status column is expected to be `pass` for all rows.

## Verify Checker Fixes Open Scala Issues (Appendix D)

In Appendix D of the paper, we mentioned the following Scala issues:

No.	File Name	Link
#9312	t9312.scala	<a href="https://github.com/scala/bug/issues/9312">https://github.com/scala/bug/issues/9312</a>
#9115	t9115.scala	<a href="https://github.com/scala/bug/issues/9115">https://github.com/scala/bug/issues/9115</a>
#9261	t9261.scala	<a href="https://github.com/scala/bug/issues/9261">https://github.com/scala/bug/issues/9261</a>
#5366	t5366.scala	<a href="https://github.com/scala/bug/issues/5366">https://github.com/scala/bug/issues/5366</a>
#9360	t9360.scala	<a href="https://github.com/scala/bug/issues/9360">https://github.com/scala/bug/issues/9360</a>
#16152	i16152.scala	<a href="https://github.com/lampepfl/dotty/issues/16152">https://github.com/lampepfl/dotty/issues/16152</a>
#9176	i9176.scala	<a href="https://github.com/lampepfl/dotty/issues/9176">https://github.com/lampepfl/dotty/issues/9176</a>
#11262	i11262.scala	<a href="https://github.com/lampepfl/dotty/issues/11262">https://github.com/lampepfl/dotty/issues/11262</a>

The test files can be found in the directory `/home/issues`. We expect all the test cases to be rejected by the checker. It can be verified by the following command:

```
for f in /home/issues/*; do
  echo "$f"
  /home/dotty/bin/scalac -Ysafe-init-global -d /home/tmp "$f"
done
```

## Reproduce the Case Study (Section 6)

To check Dotty, the Scala3 compiler, we need to first patch the compiler:

```
cd /home/dotty && git apply /home/dotty.patch
```

Now run the following commands:

```
sbt scala3-compiler-bootstrapped/clean
sbt scala3-compiler-bootstrapped/compile
```

The last command above is expected to produce 52 warnings. The warnings include:

- The 4 problems of initialization-time irrelevance described in Section 6.1, and
- One violation of partial ordering discussed in Section 6.2.

**Note:** *The initialization-time irrelevance problem between two specific objects can expose itself as several warnings if there are violations in multiple places.*

Due to code change in the Dotty repo and the improvement in implementation, the checker manages to find more violations of the two principles:

- 2 more violations of initialization-time irrelevance: `NoSymbol` / `NoDenotation` and `NameKinds` / `AvoidNameKind`.
- 2 more violations of partial ordering: `untpd` -> `Trees` -> `untpd` and `Types` -> `Names` -> `Types`.

## Implementation

The implementation is integrated in Dotty, and is located mainly in the following source files:

```
/home/dotty/compiler/library/src/scala/annotation/init.scala
/home/dotty/compiler/src/dotty/tools/dotc/transform/init/Objects.scala
/home/dotty/compiler/src/dotty/tools/dotc/transform/init/Cache.scala
```