Scala Center updates
Q2 2021 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Vincenzo Bazzucchi, 100%; Fengyun Liu, 100% until June 11th; Adrien Piquerez, 100%; Meriam Lachkar, 100%; Katja Goltsova, part-time intern; Maxime Kjaer, part-time intern; Sébastien Doeraene, 100%; Darja Jovanovic, 100%, Vincent Derouand, 40%; Valérie Pedroni since April 15th; VirtusLab team: Tomasz Godzik, 100%; Wojciech Mazur, 100%.

At a glance

MOOCs

@julienrf @vincenzobaz

Online courses have been updated to Scala 3

We have updated and published most of our online courses for Scala 3.

The courses that require Spark (Big Data Analysis with Scala and Spark, and the Scala Capstone Project) were ported to Scala 3 although only the Scala 2 remains published because Scala 2.13 support is not stable in Spark yet. We will update them as soon as Scala 3 will be correctly supported in Spark.

We have created a copy of the Scala 2 version of the courses, so that both versions (Scala 2 and Scala 3) are available for some time.

We have published a blog article that provides the links to the Scala 2 and Scala 3 versions of the courses.

New course has been published: Effective Programming in Scala

Effective Programming in Scala is a new course that teaches programmers what they need to be able to work in Scala. Since the course uses Scala 3, it can also be used to update the skills of Scala 2 programmers willing to transition to Scala 3.

Find out more in the blog article.

Maintenance

- We improved debugging of student issues by including version information in both the graders deployed to Coursera's infrastructure and handouts distributed to students
- We upgraded all the tutorials in the Scala 3 courses.

Scala documentation website

@julienrf @vincenzobaz

We have been reviewing, discussing, and merging pull requests from the community on the Scala documentation website.

Scala 3 book

We have reviewed the content of the Scala 3 book, and performed some improvements. #2062 #2073 #2076 #2079 #2088

Reflection guide

We enriched the reflection guide with what learned with the Scio project migration

Library author guide

We have updated the library author guide to use Github Actions as a CI (instead of Travis), and to use a more recent version of Scala. #2056

Library documentation workflow

We started a new project to investigate and document the best solution for library authors to publish a website with static content and API docs. We reported a bug to paradox, experimented with Docusaurus and we are currently exploring the capabilities of the new Scaladoc although it is not yet extensively documented.

Scaladoc

We have reviewed, discussed, and merged pull requests documenting the tool scaladoc for Scala 3.

Migration Guide

@adpi2 @vincenzobaz @MaximeKjaer

We gave the Scala 3 Migration Guide its final shape and moved it to the docs.scala-lang website.

Scio project

We have opened PRs for the blockers identified by the Spotify team. This allowed us to enrich the Reflection guide and to experiment with the new macro system.

Let's Talk About Scala 3 Video Series

@vincenzobaz @anatoliykmetyuk @mlachkar @julienrf @sjrd

We scripted and filmed the follow videos:

- Sébastien Doraene: Best features of Scala 3 for Scala.js (aired on April 20th)
- Julien Richard-Foy: Follow the recommended version scheme (aired on May 10th)
- Meriam Lachkar: the road to Scala 3 using scala3-migrate (aired on June 3rd)
- Anatolii Kmetiuk: the Singapor Stack (title to be refined, not aired yet)

We have also coordinated the release of a second video by Eric Loots: Exploring Coursier's Functionality Resolving Artifacts, aired on June 17th.

ScalaCon 2021

@bishabosha @adpi2 @sjrd

We gave several talks at ScalaCon 2021:

- Taste the Difference With Scala 3: how to migrate from Scala 2 to Scala 3, using the migration guide, Scala3-migrate, and interdependencies between Scala 2 and 3.
- Hands-on full-stack Scala 3 application: a tour of a complete full-stack application written using Scala 3 and its ecosystem (of Scala 2 libraries and Scala 3 ones alike).
- Fixing the three hardest library bugs of Scala.js: an in-depth tour of how we fixed parseFloat, java.util.Formatter for doubles, and java.util.regex.

Scala 3 Compiler improvements

@bishabosha @liufengyun @adpi2, and the rest of the team

With the release of Scala 3.0.0, we have focused our efforts towards further stabilizing the compiler and its compatibility with Scala 2.

We organized two internal sprees, jointly with the Scala 3 team at LAMP, in order to better share internal knowledge of the compiler codebase. We fixed some issues in the process.

Highlights

- More nuanced initialization checking of enums with the `-Ysafe-init` compiler option
- Fix the position of `end` markers and store them in SemanticDB, so that IDEs handle them better
- Other fixes in SemanticDB
- Prevent export forwarders in Scala.js within JS native types, and JS non-native traits
- Various fixes in incremental compilation
- Improvements to the experimental initialization checker

TASTy Reader For Scala 2

@bishabosha

In this quarter, we prepared the Scala 2 for the release of Scala 3, updating the TASTy Reader with the capability to read and compile against TASTy produced by Scala 3.0.0. The changes were released in Scala 2.13.6

Highlights

- Support the `INVISIBLE` TASTy flag, which hides symbols from type checking.
- Addition of new erasure modes for Scala 3 types, including "intersection type greatest lower bound" and "array element type specialization".
- Addition of a new `SCALA3X` symbol flag, optimizing speed of checks exclusive to Scala 3 definitions.
- Overhaul of the TASTy testing framework to sandbox the Scala 3 compiler from Scala 2 classes, and to enable testing experimental features of Scala 3.

Scala3-migrate

@mlachkar

We brought several improvements and bug fixes to Scala3-migrate, following feedback from the community during these first months of its existence. Notably:

- Support for migrating Test and custom configurations for projects
- Support for the final release of Scala 3
- Support for Scala.js projects
- Support for the new syntax introduce by sbt 1.5.x

sbt

@adpi2

We are continuously helping on fixing bugs in sbt 1.5.x and reviewing contributions around the BSP server.

Scala Maven Plugin

@adpi2

We assisted the maintainers and contributors of the Scala Maven Plugin in adding support for Scala 3 (released version 4.5.x).

Scala Debug Adapter

@adpi2

We fixed the display of object fields in the debugger context and we are working on showing the variable names taken from the sources instead of the class files. We are testing the debugger on Scala 3 and we will fix any issue along the way.

Metals

@tgodzik

We published the new Metals releases v0.10.2, v0.10.3 and v0.10.4 with support for Scala 3.0.0 and numerous fixes. The main new features are:

- Automatic imports for Scala 3

- Code action "Insert inferred type" for Scala 3
- `-Xsource:3` support
- Code actions "Extract and rename class"

scalameta

@tgodzik

We have stabilized the support of Scala 3 in scalameta. The latest release is v4.4.22.

- We fixed multiple issues in the parser for Scala 3
- Support for `-Xsource:3` in semanticdb plugin
- Support for Scala 2.12.14

Scalafmt

@tgodzik

Likewise, we stabilized the support for Scala 3 in scalafmt. The latest release candidate is v3.0.0-RC5. It contains fixes for formatting Scala 3 code, including a number of Scala 3-specific rewrites (done by @kitbellew) such as:

- Convert to new Scala 3 syntax
- Automatic insertion of end markers
- Drop optional braces

Scalafix

@mlachkar

We released Scalafix version 0.9.28, featuring initial support for Scala 3 (release notes for Scalafix and for sbt-scalafix).

We updated the Scalafix template to allow rule authors to write Scala 3 rules. We also updated our infrastructure with basic support for built-in rules in Scala 3, and for testing rules on Scala 3 code.

Scastie

@vincenzobaz

We brought two important upgrades to Scastie. First of all we updated to sbt 1.5.x (both for Scastie's own build and for the snippets it builds. Then we added a Scala version drop down menu to allow users to choose the version of Scala 3 that they want to use, the previous behavior was to only allow the most recent development snapshot of Dotty.

Scaladex

@adpi2

We coordinated with Sonatype to automate the indexing of all Scala 3 artifacts in Scaladex.

Standardization of semantic versioning in the ecosystem

@julienrf

We have published an sbt plugin, sbt-version-policy, which helps library authors to follow the versioning scheme Early SemVer (which is described here). The plugin:

- Declares in the build definition that the library uses the versioning scheme Early SemVer
- Provides a task intended to be run in the CI that checks that no binary incompatibility or source incompatibility is unintentionally introduced
- Provides a task intended to be run during the release process that checks that the release version is correct with respect to the compatibility guarantees of the release

We wrote a blog post (co-authored with Eugene Yokota) to explain in more detail why this should contribute to having a more robust ecosystem of Scala libraries. We also published a video (see below) showing how to use the plugin.

TASTy manipulation library

@cache-nez

We finished support for unpickling trees, and are well under way to unpickling types.

Scala.js

@sjrd

We published Scala.js 1.5.1 and later v1.6.0. Highlights include fixes to several floating-point-related functions and primitives, notably `parseFloat` and the `%e`, `%f` and `%g` conversions of `java.util.Formatter` (also used by the Scala `f"..."` interpolator). These fixes improve portability between Scala/JVM and Scala.js. Scala.js 1.6.0 introduces ``js.`import`.meta``, a new primitive to read ECMAScript 2021's meta-property `import.meta`.

We have also completely rewritten the implementation of `java.util.regex.*` to be compatible with Scala/JVM. This fixes several compatibility issues with regexes that have been reported over the past 7 years. These changes are still under review, and will likely be part of Scala.js 1.7.0.

Scala Native

@wojciechmazur

We have implemented basic Windows support and provided system-specific implementation of the Java standard library. We also have started working on a multithreaded execution runtime.

Highlights

Windows support

We have redesigned our LLVM IR code generation process to be compatible with the MSVC compiler used on the Windows platform. Garbage collector implementations were adjusted to handle portability issues. Parts of the Java standard library API were reimplemented using Windows API, beside the existing POSIX based implementation, without any additional runtime overhead.

Multithreading support

We have started research on providing multithreaded execution, reimplementing the Java Fork/Join execution model and providing proper implementation for JVM synchronization mechanisms. Current work focuses on providing a stable implementation of the concurrent execution model, while using third-party concurrency-aware, yet not performent enough garbage collectors.

Google Summer of Code

@vincenzobaz, @darjutak

We promoted our participation to GSoC via Twitter, our Blog and the other actors involved in this edition.

Together with our candidate mentors, we reviewed students' proposals. After receiving the number of slots (4) by Google, we decided which proposals to retain and we completed the necessary paperwork. Finally we welcomed students and mentors in a dedicated Slack space and we are tracking the evolution of the projects.

Communication and Management

@darjutak

We are incredibly happy to have participated in the Scala 3 release and overwhelmed by the positive feedback and excitement from the community, some of it visible here: https://twitter.com/scala_lang/status/1393139114535444484.

At the same time, we were quite saddened to learn about a community member's horrifying experience in the Scala community, and we are determined to do all that is in our power and area of influence to do the necessary to prevent such things from happening again. We created a "Scala Community Management and Governance" strategy proposal and we want to start the work as soon as possible.

- **Personnel movements**

  Fengyun Liu accepted a job with OracleLabs after finishing his mission at EPFL and Scala Center on June 11th 2021.

  Valérie Pedroni started her 1 year internship mission on April 15th 2021.

- **New Advisory board member**

  Databricks officially joins June 17th 2021

- **Published 4 blog posts:**

  Effective Programming in Scala

  The Scala Center Action Towards a More Safe and Respectful Community Environment

Scala 3 in sbt 1.5

- Continued filming and publishing “Let’s talk about Scala 3” video series

  Best Features of Scala 3 for Scala.js, by Sébastien Doeraene, Scala Center

  Follow the Recommended Versioning Scheme, by Julien Richard-Foy

  The road to Scala 3 using scala3-migrate, by Meriam Lachkar, Scala Center

  Exploring Coursier's Functionality Resolving Artifacts, by by Eric Loots, Lunatech

- **Participated and presented in online conferences:**

  Scala 3 Release party, leading the panel, see more here.

  ScalaCon (videos will be available in Nov/Dec 2021)

  Hands-on full-stack Scala 3 application, by Adrien Piquerez

  Fixing the 3 hardest library bugs of Scala.js, by Sébastien Doeraene

  Taste the difference with Scala 3: Migrating the ecosystem and more, by Jamie Thompson

- **Communication team produced:**

  Published followup on 2nd Organizers Summit

  5-year report concept (more in the next quarter)

  Marketing campaign for MOOCs promotions, emphasis on the new Effective Scala course

  Internal onboarding documents

  Various support of the Scala Center members and ongoing activities

- Created the “Scala Community Management and Governance” strategy proposal

Community monthly updates and project roadmaps are publicly visible here: https://contributors.scala-lang.org/c/scala-center/25