

Scala Center updates

Q4 2020 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Vincenzo Bazzucchi, 100%; Adrien Piquerez, 100%; Meriam Lachkar, 100%; Vincent Derouand 40%; Valérie Pedroni intern; Sébastien Doeraene, 100%; Darja Jovanovic, 100%; VirtusLab team: Tomasz Godzik, 100%; Wojciech Mazur, 100%.

External contractors, part time: Adam Goodman, leadership and governance expert, Gordon Savicic, designer, Tanya Murphy, impact report expert.

At a glance

[At a glance](#)

[Education & Documentation](#)

[MOOCs](#)

[Scala 3 documentation](#)

[Scala at Universities](#)

[Scala Website Investigation and Modernisation](#)

[External Communication](#)

[Let's Talk about Scala 3 Video Series](#)

[Scala Italy](#)

[Scalabase](#)

[ScalaCon](#)

[LinkedIn Page](#)

[Advent of Code](#)

[Scala Online Shop](#)

[Mapping Out Companies](#)

[Scala Courses Promotion](#)

[Community](#)

[Inclusive Language Guide \(SCP-025\)](#)

[Compiler](#)

[TASTy Reader](#)

[Scala 3 Support in Scala Native](#)

[Energy Efficiency of Scala](#)

[Scala.js](#)

[Scala 3 Compiler Maintenance](#)

[Ecosystem](#)

[Metals](#)

[Bloop](#)

[Scaladex](#)

[Coursier](#)

[Sbt](#)

[Scala Debug Adapter](#)

[Management](#)

[Communication strategy](#)

Commitment to Scala 2

During the last Advisory Board meeting, concerns were raised about the feeling that the Scala Center was focusing too much on Scala 3, to the detriment of Scala 2 users. Therefore, we examined our past and existing involvement, discussed with Lightbend, and redefined our commitment to supporting the Scala 2 user base.

We are committed to supporting the users of Scala 2 as much as the users of Scala 3, in the areas of tooling, community, documentation, and (to some extent) teaching. The support of the language and compiler for Scala 2 is left to the care of Lightbend, as it has been since the creation of the Scala Center.

During the year surrounding the release of Scala 3, our team was heavily focused on Scala 3. We actively coordinated all the efforts leading up to the release, and we continued to dedicate significant efforts after the release for the tooling and documentation to catch up with that of Scala 2. For the past 6 months, the situation having somewhat stabilized, we have gone back to more regular development, which typically applies equally to Scala 2 and Scala 3.

Starting this quarter, we are taking more deliberate steps to clarify our commitment to Scala 2. In particular, our reports (including this one and our posts on contributors) clearly show what major version(s) of Scala they apply to. Each technical section of this report has a notice "for Scala 2", "for Scala 3", or "for Scala 2 and 3".

Education & Documentation

MOOCs

@julienrf, @vincenzobaz, for Scala 3.

We listened to the learners and we worked on various improvements to the courses [Effective Programming in Scala](#) and [Functional Programming Principles in Scala](#):

- New lessons on literal functions and exceptions,
- More actionable feedback in automated graders,
- 43 additional quizzes.

Scala 3 documentation

@julienrf

We identified several issues in the Scala documentation website (summarized [here](#)), and we addressed them:

- We simplify the path to get started by unifying the two “Getting Started” pages for Scala 2 and Scala 3 into a single page that supports both versions of Scala ([scala/docs.scala-lang#2197](#)),
- We added a search bar on every page ([scala/docs.scala-lang#2201](#)),
- We clarified the structure of the navigation menus ([scala/docs.scala-lang#2206](#)).

Scala at Universities

@vincenzobaz

As part of our education mission, we want to promote the usage of Scala as a vehicle to teach programming and software engineering. We are currently running a study to identify the factors that lead professors to use Scala.

We interviewed 9 university professors. We collected their feedback on the advantages of using Scala for teaching and on the main difficulties when doing so. Many elements are reported by multiple teachers such as tooling friction, lack of teaching resources, and set up process. Similarly, most teachers agree that the language has many properties that are beneficial for teaching (regularity, multi-paradigm, etc.).

In December, we interviewed Romain Edelman who is working on introducing Computer Science education in Vaud’s High School system, and offered some interesting insight on the software and hardware tooling he is considering for this project.

Scala Website Investigation and Modernisation

@bishabosha, for Scala 2 and Scala 3

We conducted an investigation into the websites of other programming languages, comparing the beginner experience with that of [scala-lang.org](#), and the impact that could be delivered in the future.

We presented a short summary of the initial findings at the previous Advisory Board meeting. Following that, we created another meeting with members of Lightbend, VirtusLab, and 47 Degrees to present our mission statement for the website going forward, and to secure the trust from attendees that the Scala Center will be responsible for proposing changes to fulfill this mission and ensuring that they are implemented. We then produced a further [web presentation](#) about the state of programming language websites and some immediate suggestions for Scala’s own website.

External Communication

Let's Talk about Scala 3 Video Series

@julienrf, @vincenzobaz, @adpi2, for Scala 3

[Series of short videos](#) about Scala 3, covering a variety of themes like how to get started, how to take advantage of the new language features, or how to migrate from Scala 2.

(for Scala 2 and Scala 3) We published a video [Using sbt Server for a faster development workflow](#), explaining how to leverage sbt server in the day-to-day workflow.

We also wrote the script and filmed a new video about [Program Entry Points](#) in Scala 3. The video is currently being edited and should be published in early 2022.

Scala Italy

@vincenzobaz

[Scala Italy](#) happened online on the 23rd of October.

We gave a talk, in Italian, demonstrating how to write a full-stack application in Scala 3.

Scalabase

@adpi2, @vincenzobaz, @bishabosha, @mlachkar

[Scalabase](#) is a conference for developers in the early stages of their Scala journey. It happened online on the 28th of October.

We ran an “install party” to assist new users in installing the required tools to program in Scala, and then we ran a workshop guiding the attendees in the implementation of a Snake Game. You can find the material [here](#).

ScalaCon

@mlachkar @tgodzik

[ScalaCon](#) is a series of virtual conferences designed to bring the Scala community closer together.

We gave the following talks:

- [Boost your productivity with Scala tooling!](#)

- [Metals—Your IDE for Scala 3](#).

LinkedIn Page

@valeriePe

We created a [LinkedIn page](#) for the Scala Center organization where we publish posts every week. For this quarter 15 posts were published.

We met our initial 200 followers goal in one week, with 273 followers. Our engagement rate is fluctuating between 5 and 9% (compared to the 2% engagement rate that's considered as good). At the beginning of 2022, we have 410 followers.

Advent of Code

@adpi2, @mlachkar, @vincenzobaz, @bishabosha, @sjrd, @julienrf, @tgodzik, for Scala 3.

We created a [website](#) to share our solutions to the [Advent of Code](#) puzzles. We also gathered more than 60 solutions from community members.

Scala Online Shop

@valeriePe

We want to create a Scala online shop to **offer a range of Scala merch to community members while giving them the possibility to donate**. Potential buyers could buy a product at its original price or decide to donate more money to the Center and buy the same product at a higher price. Donations will be encouraged by selling special products such as signed books or limited articles.

This platform would participate in raising funds towards the Scala Center and allow Scala fans and community members to support the center while highlighting their affiliations to Scala language with exclusive merchandise.

To achieve that:

- We had a meeting with the [EPFL shop](#) people to discuss the possibility of a Scala Online Shop.
- We agreed to sell our products through their website and use their infrastructure to ease the process.
- Since then we have worked in close collaboration with them to set up the shop, order our first products and agree on logistics.

We expect to sell our first products in the first quarter of 2022.

Mapping Out Companies

@valeriePe

In an effort to ease the fundraising process and have a better idea of our potential affiliate or AB members we created a detailed map of ~200 companies using Scala. In 2022 we plan on expanding the document, introducing foundations and other possible funding opportunities and use it to reach out from mid of this year.

Scala Courses Promotion

@valeriePe, @julienrf, @vincenzobaz

In collaboration with Coursera and 47 Degrees, we recorded two video trailers presenting the MOOCs of the Scala Center, and we co-created visuals that we will use for a communication campaign. The campaign is expected to start in Q1 2022.

Community

Inclusive Language Guide (SCP-025)

@sjrd @darjutak @valeriePe

We finished the inclusive language guide [requested by this proposal](#). In the past quarter, we gathered feedback from key parties involved in the development of Scala and adapted the guide accordingly. It is now published at <https://www.scala-lang.org/contribute/inclusive-language/>.

We have applied it to many of our repositories, although it is possible that we missed some, especially the least active ones. We are collaborating with the LAMP team to apply it to the dotty repository, which requires some work to adapt the CI and benchmarking infrastructure.

Compiler

TASTy Reader

@bishabosha, for Scala 2 in the context of Scala 3

The TASTy reader allows Scala 2.13 programs to use libraries compiled with Scala 3. The release of Scala 3.1.0 introduced a new version of the TASTy format, which is not supported by Scala 2.13.6.

We have updated the implementation of the TASTy reader in the Scala 2.13 compiler to support Scala 3.1.0 ([scala/scala#9791](#)). This has been released in [Scala 2.13.7](#).

Scala 3 Support in Scala Native

@wojciech, for Scala 3

We implemented initial support for the generation of NIR files (ie, Scala Native binaries) as a Scala 3 compiler plugin, and for test frameworks (JUnit and reflective instantiation).

We ported the Scala Native compiler plugin to Scala 3, which now covers all functionalities known from Scala Native for Scala 2. We researched an additional compiler phase that would adapt the Scala 3 LazyVals implementation, by rewriting unsupported reflective calls to Scala Native intrinsic operations. We rewrote the Scala Native sbt build to allow for safe cross-compilation for all binary versions of Scala and we are currently testing and stabilizing Scala 3 support.

Energy Efficiency of Scala

@julienrf, @wojciech, for Scala 2 and Scala 3

This project assesses the energy efficiency of Scala programs, based on the article "[Ranking programming languages by energy efficiency](#)".

We published a [blog article](#) comparing the energy efficiency of Scala programs with programs written in other programming languages.

Scala.js

@sjrd, for Scala 2 and Scala 3

We released [Scala.js 1.8.0](#).

A change in the default configuration of Node.js 17 caused [a critical incompatibility](#) with Scala.js. We immediately published a workaround, soon followed by a proper fix. Users of Scala.js who would like to use Node.js 17 are encouraged to upgrade to Scala.js 1.8.0, which contains the fix.

We also addressed a bad interaction with sbt's client/server mode, which was released as part of Scala.js 1.8.0.

Scala 3 Compiler Maintenance

@bishabosha, for Scala 3

We fixed a bug that caused enum definitions to type check differently depending on whether its companion object was declared before or after the enum in a file. This was caused by checking too early for inheritance of `java.lang.Enum`, required for synthesis of the ordinal method. We moved the synthesis to after type checking, and improved the content of the error message for when the user tries to implement ordinal themselves. See the PR [#13952](#) for more details.

We merged support for incremental compilation of inline methods that refer to other inline definitions in their body (PR [#12931](#)).

We also merged a fix for a bug that crashed the compiler when the `-sourcepath` compiler flag is set and incremental compilation reaches a non-inline definition that calls an inline definition (PR #14050).`

We fixed a bug introduced in Scala 3.1.0 (summoning a ``deriving.Mirror.SumOf[T]``, where ``T`` is a hierarchical sum type compiled by Scala 3.0.x, could cause a `ClassCastException`). To fix this we introduced more checks in the compiler before summoning Sum Mirrors (PR [#14035](#)).

Ecosystem

Metals

@tgodzik, for Scala 2 and Scala 3

(for Scala 3) We worked a bit on the Scala 3 compiler to improve user experience when dealing with annotations. Firstly, we added annotations to semanticdb, which can be later used to see what method/class is annotated. We also worked on making sure that completions and hovers can work with incomplete or standalone annotations. This should greatly improve user experience in Metals and other tools.

In Metals itself, we rewrote the sbt-metals plugin so that it fully relies on the official SemanticdbPlugin from sbt ([scalameta/metals#3200 3](#)) without custom support in the plugin itself.

We also released two new Metals version

<https://scalameta.org/metals/blog/2021/10/26/tungsten 2> and <https://scalameta.org/metals/blog/2021/11/03/tungsten 2> which bring in a lot of new features.

(for Scala 2) One of the most prominent new features was the expression evaluator for Scala 2, which greatly improves the debugging experience for the users. This effort was done by <https://github.com/tdudzik> with big help from Scala Center itself. We also added type on selection for Scala 2, which was done by <https://github.com/KacperFKorban>, and lets users check types of any expression they select. We've also introduced support for new Scala versions 3.1.0 and 2.13.7 as well as greatly improved the support for Scala 3 with numerous fixes done mostly by <https://github.com/dos65>. This release had some new features from external contributors such as searching for text in dependencies (<https://github.com/Z1kkurat>) or a file analyzer that is capable of translating Scala/Java files to a number of different formats such as semanticdb, javap, tasty or decompiled code (<https://github.com/Arthurm1>).

Bloop

@tgodzik, for Scala 2 and Scala 3

Bloop is seeing continuous improvements over the two new versions that has been recently released <https://github.com/scalacenter/bloop/releases/tag/v1.4.10> and <https://github.com/scalacenter/bloop/releases/tag/v1.4.11> :

These releases included numerous improvements for Maven integration, which seems to be used by a number of people, as evidenced by the numerous issue reports we were getting. With the release of Scala 2.13.7, it turned out that we needed to do a small fix in order to support it in Bloop, which also brings to light the importance of switching back to a normal fork (work is finishing). Bloop has also seen fixes and features for outside contributors such as Gradle improvements and an option to generate semanticdb for Java files used currently by Metals (both <https://github.com/ArthurM1>).

Scaladex

@adpi2, @mlachkar, for Scala 2 and Scala 3

[Scaladex](#) is an index website for all Scala libraries. The project suffered from usability and reliability issues, which we have been addressing.

We made the underlying infrastructure more reliable by overhauling the persistent storage system. We improved the performances ([scalacenter/scaladex#729](#), [scalacenter/scaladex#723](#)), and made the data more reliable by reworking the github synchronization ([scalacenter/scaladex#743](#)), and Elasticsearch synchronization ([scalacenter/scaladex#742](#)).

We successfully deployed the project to production after this major rewrite. Overall, the refactoring has strengthened the foundations of Scaladex and prepared the ground for new features and external contributions.

Coursier

@sjrd, @adpi2, for Scala 2 and Scala 3

Coursier can be used as an installer for Scala. However, it was [reported by the Advisory Board](#) that it suffers from several issues that make it unsuitable for beginners.

We fixed the support of Scala 3 artifacts resolution ([coursier#2194](#)). Based on foundational work by Alexandre Archambault, we added support for installing an application exclusively from a zip file ([coursier#2193](#)) so that coursier can install the official sbt script and launcher. We proposed a technical solution to merge Scala 2 with Scala 3 related applications ([coursier#2231](#)).

We finished all the changes to the coursier CLI that were planned [in our roadmap](#), including:

- add a feature to make sure that a given application is always installed from its official distribution (e.g., sbt),
- add a feature to override application settings based on the version, which effectively allows merging Scala 2 and Scala 3 into a single application,
- adjust the documentation and help menus to make them more beginner-friendly, focused on the getting started experience.

These changes are currently available in [coursier v2.1.0-M1](#). Once a final release is cut (see the [tracking issue](#)), we will update the official [scala-lang.org](#) website to recommend coursier as the default installer everywhere, for Scala 2 and for Scala 3.

sbt

@adpi2, for Scala 2 and Scala 3

We added support for sources in the base directory in sbt BSP implementation ([sbt#6701](#)).

In order to prepare the release of sbt 1.6, we fixed a few bugs with sbt server ([#6547](#), [#6698](#)), affecting all versions of Scala.

Scala Debug Adapter

@adpi2, for Scala 2 and Scala 3

After the release of expression evaluation in Scala 2.12 and Scala 2.13, we fixed a number of bugs in the debug adapter itself and its integration with Bloop and Metals. Also, we added support for Java 17 ([scalacenter/scala-debug-adapter#115](#)).

Management

@darjutak

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

- Anatolii Kmetiuk joining the team, onboarding started in December
- Vincenzo Bazzucchi announced he is leaving the team in March
- IC EPFL granted 98K CHF from towards the Governance and other strategic projects
- Thanks to this support, we signed contracts with external experts for: Community Management and Governance, 5-year impact report)

Communication strategy

@VincentDrnd

A first communication strategy for the Scala Center was discussed with Darja in November. Communication goals are divided into 5 strategic areas:



1. Image & reputation
 - a. Improve Scala Center image and reputation
 - b. Unify Scala identity
2. Projects & core activity
 - a. Promote Scala Center developments and education outcomes
3. Community & governance
 - a. Gain and retain trust relationship between Scala Center, Stakeholders and Community
 - b. Actively participate to the extension of the Scala Community
 - c. Develop Corporate communication
4. Business
 - a. Support growth of the Business sector
5. Funding
 - a. Support growth of revenues
 - b. Promote Scala Boutique

The team was involved in this process, as there was a first presentation in early December, followed by a first workshop on the 15th. Strategy and good practices were at the heart of the discussions aiming to develop a more efficient way to communicate by everyone and identify what's needed by the team (annual communication plan, guidelines, tools, documentation, etc.). The second workshop is planned for mid-January.