

Scala Center updates

Q4 2020 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Vincenzo Bazzucchi, 100%; Fengyun Liu, 100%; Adrien Piquerez, 100%; Meriam Lachkar, 100%; Katja Goltsova, part-time intern; Sébastien Doeraene, 100%; Darja Jovanovic, 100%; VirtusLab team: Tomasz Godzik, 100%; Wojciech Mazur, 100%.

External contractors, part time: Noel Welch, Alvin Alexander, Ólafur Geirsson.

At a glance

- [At a glance](#)
- [MOOCs](#)
- [Scala 3 documentation](#)
- [Google Summer of Code](#)
- [Let's Talk about Scala 3 Video Series](#)
- [Scala 3 Migration Guide](#)
- [Scala 3 support in sbt](#)
- [Mill](#)
- [Scala Debug Adapter](#)
- [Scaladex](#)
- [TASTy Reader For Scala 2](#)
- [Scala 3 Compiler improvements](#)
- [Scala3-migrate](#)
- [Scalafix](#)
- [Scalafmt and scalameta](#)
- [Metals](#)
- [Scala Native](#)
- [Scala.js](#)
- [Communication](#)
- [Management](#)

MOOCs

@julienrf @vincenzobaz

Effective Scala course

We have been improving the slides of the course, following feedback from the beta tests. In addition, we have been recording the lectures for Effective Scala, and are almost done. For the

last ones, we are waiting for the Scala 3 syntax to be frozen. The assignments are all completed and are on Coursera, including the automatic graders.

Other courses

We have updated to the latest Coursera SDK for all the existing courses. This was a mandatory update for the courses to keep running on the infrastructure in 2021.

We have also internally migrated all the existing courses to Scala 3.0.0-M3, using updated dependencies and Scala 3 idioms: enums, indentation-based syntax, contextual abstractions and top-level definitions. For the release, we are blocked by Scala 2.13 support in Spark, which should happen soon (<https://issues.apache.org/jira/browse/SPARK-25075>).

Scala 3 documentation

@julienrf @vincenzobaz

We coordinated the integration of Scala 3 documentation into docs.scala-lang.org with LAMP.

Google Summer of Code

@vincenzobaz @darjutak

We aim to coordinate the 2021 edition of GSOC for the Scala organization. We are collecting as much knowledge as possible from previous editions, in particular from mentors, students and organizers. Thanks to this precious information we are elaborating a strategy to have a great 2021 edition. We are also getting in touch with the first batch of mentors.

Let's Talk about Scala 3 Video Series

@darjutak @julienrf @vincenzobaz @bishabosha

We are preparing several videos around the release of Scala 3, in a partnership with 47 Degrees. Two videos are already in the pipeline:

- Getting started with a Scala development environment, using Coursier, by Eric Loots.
- Breaking down the high level intent to make a guessing game into its smallest parts, coding the parts separately and putting together the pieces. The video will demonstrate using the simple parts of Scala 3, and is aimed at newcomers.

Scala 3 Migration Guide

@adpi2 @mlachkar

We have added an sbt tutorial that explains the steps required to [port an sbt project to Scala 3](#): identifying the project, preparing the dependencies, setting up cross-building, configuring the Scala 3 compiler, solving the incompatibilities. This tutorial still relies on the sbt-dotty plugin. It will soon be translated into sbt 1.5.0, which features direct Scala 3 support (see below).

We have also added information about [migrating compiler options](#), since Scala 3 does not have exactly the same set of options as Scala 2.

Scala 3 support in sbt

@adpi2

For better support of Scala 3 in sbt, we bumped the version of Zinc to 1.4.x in the Scala 3 compiler bridge. This change improves the analysis report produced by Zinc when compiling Scala 3. It will also enable the use of remote caching and pipelining in Scala 3, although those features have not yet been tested with Scala 3.

We have integrated the main functionalities provided by the sbt-dotty plugin in the default settings of sbt, sometimes in a different form.

We have released the added support in [sbt 1.5.0-M1](#).

By default, sbt is now able to manage the instances of Scala 3, that is, the compiler and libraries. It is also able to manage Scala 3 dependencies and to compile Scala 3 sources. From 3.0.0 onward the Scala binary version will be '3', thus implying the bidirectional compatibility between any two future minor versions.

We replaced the `isDottyCompat` extension method by the static `for3Use2_13` object whose usage is:

```
("a" % "b" % "1.0").cross(CrossVersion.for3Use2_13)
```

Symmetrically we added the `for2_13Use3` that has no counterpart in sbt-dotty.

For now, the `doc` task is made available on Scala 3 by setting:

```
thisBuild / resolvers += Resolver.JCenterRepository
```

This is due to `scala3doc` depending on `Dokka`, whose artifacts are not available in the default resolvers. This dependency is planned to be removed.

Mill

@adpi2

We bumped the version of Zinc to 1.4.x in Mill. The underlying plan is to move the Zinc ecosystem away from the deprecated APIs of Zinc before Scala 3 is released.

Scala Debug Adapter

@adpi2

The [scala-debug-adapter](#) is a server-side implementation of Microsoft's [Debug Adapter Protocol](#) for the Scala language. We extracted it from Bloop so that we can reuse it in other build tools of the ecosystem, namely sbt and Mill. As part of this project, we are working on the sbt-debug-adapter, an sbt plugin that augments the sbt server with the debug capabilities. As soon as released, it will be used for debugging inside Metals. None of that project has yet been released.

Scaladex

@adpi2 @bishabosha

We make sure that Scaladex is up-to-date with the new versions of Scala 3, Scala Native and Scala.js, and that all the new library versions are indexed.

We released the Version Matrix page to help with the migration on all platforms.

TASTy Reader For Scala 2

@bishabosha

In this quarter we have continued to maintain the TASTy reader after its release in Scala 2.13.4, adding support for the latest Scala 3 public releases.

Highlights

- A first version of the TASTy Reader was released as part of Scala 2.13.4 on November 19th.
- We published a blog explaining benefits of TASTy Reader: <https://www.scala-lang.org/blog/2020/11/19/scala-3-forward-compat.html>
- Reading TASTy produced by Scala 3.0.0-M3 is supported in Scala 2.13.5 nightly releases.
- Reading opaque type aliases is now supported in Scala 2.13.5 nightly releases.

Scala 3 Compiler improvements

Scala 3 Enum Stabilization

@bishabosha

We have been improving the behavior of enums with metaprogramming facilities in Scala 3.

- We removed the deprecated `scala.Enum` type alias, in favor of the `scala.reflect.Enum` trait.
- Scala 3.0.0-M3 enables inline match to deconstruct recursive enums.
- We have work-in-progress to enable dependent typing for enums.

Scala 3 Export Clause Stabilization

@bishabosha

We have focused on improving corner cases of export clauses, with regards to incremental compilation, reflection, and correct adaption of exported forwarders.

- Scala 3.0.0-M3 adds support for export clauses in TASTy, and can capture wildcard export dependencies on classes in incremental compilation. Wildcard export from packages is now restricted.
- Correctly export inline vararg methods.
- Correctly export Java vararg methods.
- Export companions to case classes.

Scala.js support in Scala 3

@sjrd

We finished the support of Scala.js in Scala 3. Scala 3.0.0-M3 ships with complete support for Scala.js.

Scala3-migrate

@mlachkar

Scala3-migrate is a tool to migrate projects from the latest 2.13 version to the latest 3.0.0. Following the timeline and objectives we described in [the relevant Scala Contributors post](#), we have implemented the sbt-plugin with the two main rules that infer result types, type applications, implicit parameters and implicit conversions.

In concrete terms, we currently provide 3 commands, and the fourth one is in progress:

- `migrate-prepare` `projectId`: Remove dropped features and fix syntactic changes introduced with Scala 3.
- `migrate-scalacOptions` `projectId`: Examine the `scalacOptions` used in the project, and propose changes with Scala 3 settings.
- `migrate-libraries` `projectId`: Migrate the library dependencies (in progress).
- `migrate` `projectId`: Add the necessary explicit types or implicit parameters to make the project compile in Scala 3.

Scalafix

@mlachkar

We released 3 new versions of Scalafix that improve error reporting in the new API, fix a memory leak reported by multiple users, and add some functionality for rules' developers. ([0.9.23](#), [0.9.24](#), [0.9.25](#))

Scalafmt and scalameta

@tgodzik

We are improving the support for Scala 3 in scalafmt and scalameta:

- Support for almost all new Scala 3 features ([documented here](#))
- Support in the Scalameta parser for the changes in Scala 3 syntax, along with a number of fixes to parsing

Metals

@tgodzik

- Update of Metals to use the scalameta parser for Scala 3, which enables features such as document outline
- New decorations that show inferred types, implicit classes, implicit parameters and implicit conversions
- Navigation for stacktraces returned via the debug adapter protocol

The above features have been released in version 0.9.5 to 0.9.10, whose release notes can be accessed from [here](#).

TASTy manipulation library

@cache-nez

Planning for a future where TASTy is the main format on which we base distribution and compatibility of libraries, we are working on a library to read and inspect TASTy files that is independent of the compiler(s).

At this point, most of the data structures and unpickling procedures are implemented, although there is still much work left to do for it to be anywhere near usable.

Long-term, this library will be the basis of tools like an equivalent of MiMa for TASTy compatibility.

Scala 3 support in Ammonite

@alexarchambault

We have almost completed adding support for Scala 3 in Ammonite. The first half of the work, namely separating the component of Ammonite that talks to the compiler, so that the rest of Ammonite is isolated from the version of the compiler, has already been merged. The second half is in a working state, but needs to be polished before merging. This should happen within the next two weeks.

Scala Native

@wojciechmazur @sjrd

We have finally released [Scala Native 0.4.0](#). Highlights include support for Scala 2.12 and 2.13, and a better testing infrastructure. It also contains a long list of small improvements and bug fixes.

Now that we have published this version, for which we had to break binary compatibility, we will be able to publish smaller updates more often (every 2-3 months).

We have also published a new version of portable-scala-reflect with support for Scala Native 0.4.x, which will make it easier for testing frameworks to support Scala/JVM, Scala.js and Scala Native using a single codebase.

Miscellaneous

- We have optimized the build process, which resulted in a significant reduction of memory usage.
- We have extracted the partially implemented `java.time` and `java.text` packages from the core repository to allow proper third-party implementations.
- We have improved compliance of Scala Native with the JDK library implementation and Unicode specification.

Scala.js

@sjrd

We published [Scala.js 1.3.1](#) and later [Scala.js 1.4.0](#). Both releases feature significant performance improvements: the former for all code on Firefox, and the latter for Arrays on all platforms.

The major highlight of Scala.js 1.4.0 is the support for dynamic module loading, which was implemented by Tobias Schlatter. This allows an application to start without all of its components, which will be downloaded and executed only once used for the first time.

Scala.js IR interpreter (student project)

In the context of a master's thesis, Ivan Yurov, an EPFL student, developed a Scala.js IR interpreter, supporting the full range of JavaScript interoperability features of Scala.js. The implementation can be found at <https://github.com/youroff/tint>.

Communication

@darjutak

- 6 blog posts on [Scala-lang.org](#)
- Launched a “Scala 3 - Developer’s Preview Survey” (results coming soon)
- Various contributions and announcements on [Contributors](#)
- Supported by 47degrees and in collaboration with Eric Loots from Lunatech, recorded 4 short education videos topic “Let’s Talk About Scala 3” (releasing soon)
- Expecting to employ a Communications manager, 40%, 1 year mission document and standardize our practices, as well as propose a strategy

Management

@darjutak

- Q4 2020 will be known as “all hands on deck” for Scala 3-M3 (almost RC1) release. Management was focused on supporting each person’s project, and wrapping up the big coordination efforts. We’ve worked with ~41 people (+ community) on ~30 projects, and successfully released Scala 3-M3 as “Developer’s Preview”, in other words, if no major concern was raised, it would become Scala 3-RC1. We hope to see this happen very soon.
- In collaboration with EPFL we were presented with an opportunity to share a Communication manager between 2 centers (eSpace Center and Scala Center), we are currently in the hiring process hoping to have a new member starting 1st of March, 40% assignment, for 1 year mission.

- In this period we kept our internal team sprees (training) and took professional “Speaking on camera” training, as a preparation for our “Let’s Talk About Scala 3” video project.
- In Q1 2021, the management staff is focusing on designing in-depth strategies for the Scala Center’s tech, education, communication, and community departments. More on this in Q2.