DRAFT - in DISCUSSION

DataStax Drivers Donation

Status

Current state: Draft
Discussion threads:

- DISCUSS thread
- Initial discussion (pre-CEP)

JIRA: TBD
Released: Not Released

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Scope

In 2011, drivers were removed from the Apache Cassandra project. An inspection of the project history shows that in-tree drivers weren't working well "because the people who wanted to contribute to the drivers were for the most part not Committers, and the committers for the most part weren't interested in reviewing drivers patches".

In 2013, DataStax released the Java Driver and funded the creation of other driver languages. Further discussion on the topic clarifies some of the tension the project faced: "when you have six or seven or more parts of the tree whose committers' expertise does not overlap, [...] it make[s] sense for these [projects] to be organized as separate projects. [...] Making them their own projects has resulted in more and higher quality drivers overall."

Since late 2016, the Cassandra driver landscape has become fragmentary. Two main issues are commonly mentioned:

- The lack of official drivers does not favor cohesion around the Cassandra project. Many participants in the Cassandra ecosystem have created forks of the DataStax drivers but have not historically contributed changes back to the original drivers.
- The dependency of the Apache Cassandra server itself on two of the DataStax drivers (Java and Python) has implications for the project's independence. This tension has been exacerbated by the release, in 2019, of version 4.0 of the Java Driver with major

API changes. This was followed by a transition to maintenance mode of the 3.x series which is still used by the server and widely in the community.

Goals

Contributors employed by DataStax are offering to donate to the Apache Software Foundation all seven DataStax-funded drivers, currently hosted in the following GitHub repositories:

1. datastax/java-driver: DataStax Java Driver for Apache Cassandra
2. datastax/python-driver: DataStax Python Driver for Apache Cassandra
3. datastax/nodejs-driver: DataStax Node.js Driver for Apache Cassandra
4. datastax/csharp-driver: DataStax C# Driver for Apache Cassandra
5. datastax/cpp-driver: DataStax C/C++ Driver for Apache Cassandra
6. datastax/ruby-driver: DataStax Ruby Driver for Apache Cassandra
7. datastax/php-driver: DataStax PHP Driver for Apache Cassandra

We think it is important to maintain the drivers together to retain cohesive API semantics and make sure they have similar functionality and feature support. It is therefore requested that all drivers be accepted eventually; see "Approach" and "Timeline" below for practicalities on the transfer operation itself, which does not need to be an all-at-once transfer.

Approach

Transferring large pieces of software like the drivers will require a strong level of coordination and involvement from designated members of both organizations involved in this operation (DataStax contributors, legal, and the ASF).

The details of various aspects of this operation, such as governance, source code hosting, intellectual property, etc. are covered in detail in "Proposed Changes" below, where we discuss ways to deal with the challenges they pose and avoid identified potential pitfalls.

In order to minimize the risks of creating a suboptimal situation both for the drivers and the Cassandra project itself in the future, the donation process will be iterative, and will start with only the Java driver in a first phase; then, in a second phase, it will be extended to the remaining drivers.

Once the Java driver is transferred, and before the others are transferred, we will revise the methodology described in this CEP, and if necessary, revise its parameters and adjust them accordingly. A second CEP may be required if the changes to the methodology are found to be substantial.

Timeline

There are two phases to be considered:

1. Java driver donation and transfer to the ASF: We believe that this should be executed after the Cassandra 4.0 GA release, in order to not disturb the current efforts towards this major milestone.
    a. The timeline should allow for the whole intellectual property clearance process to take place, see below "Intellectual property".
2. Remaining 6 drivers donation and transfer to the ASF: TBD based on discoveries from 1.

Mailing list / Slack channels

Formal discussions on this CEP should be held on the Apache Cassandra Dev ML, as per ASF guidelines.

An initial discussion (pre-CEP) already happened on the Dev Mailing List. A formal [DISCUSS] thread has been started, as per the Apache Cassandra CEP process.

ASF Slack may also be used for informal discussions; a new #cassandra-cep-drivers-donation Slack channel will also be created to that effect.

Related JIRA tickets

No existing Apache Cassandra Jira tickets relate to this CEP. However the following ticket can be mentioned for its historical relevance:

- CASSANDRA-2761: Initial discussion around the removal of CQL drivers from the 0.8 branch.

Also, see below "Proposed Changes" for proposed new Jira projects.

Motivation

By donating all its drivers to the ASF, we hope to:

- Provide the Apache Cassandra project with "official" drivers and resolve the concern by the project community from the lack of drivers governance.
- Demonstrate community goodwill and address the ask from some Cassandra PMC members that drivers should not be controlled by any organization external to the ASF.
- Increase the cohesion of the Cassandra ecosystem by hosting together again both the Cassandra server and its most popular CQL drivers.
- Provide the Cassandra project with a client-side reference implementation of its own native protocol. The DataStax Java driver, indeed, has served so far as the de facto reference implementation of said protocol.

We should however avoid the situation we had back in 2011. In particular, this CEP attempts to strike a balance between the need for independent stewardship for both drivers and server, especially for day-to-day work; while keeping a reasonable amount of common shared governance for high-level decisions (roadmap, common features, etc.).

Audience

The donation outlined in this CEP would be beneficial to the entire Cassandra community and ecosystem.

Depending on the persona, two main audience groups can be outlined:

- Apache Cassandra committers and PMC members, as well as DataStax driver committers will likely be affected directly to some level, but hopefully such impacts will remain limited mostly to adaptation to new governing bodies and rules, and to communication channels: Jira, mailing lists, Slack, etc. – which will be monitored closely.
- Apache Cassandra users in the broad sense should benefit from the donation by having a stronger community built around the main project. However users should not be affected by the practicalities of the change proposed in this document. In particular, we would like to minimize the disruption caused by the donation and avoid massive user-facing changes to any of the drivers or server APIs, and to the drivers release funnels. See below "New or Changed Public Interfaces" for a detailed discussion.

Proposed Changes

A. Governance

We think it is best to avoid creating a separate top-level Apache project, and suggest that the drivers should be included in Apache Cassandra as a single subproject under the governance of the Apache Cassandra Committee.

There is precedent for incubating drivers as separate top-level projects: for instance Apache Curator, which is an Apache Zookeeper client donated by Netflix, is a top-level Apache project. It seems however that this dual-project approach has caused significant disruption to the projects when coordinating releases and addressing legal concerns.

On the other hand, many Apache projects have subprojects: Apache Felix and Apache Cocoon, for instance.

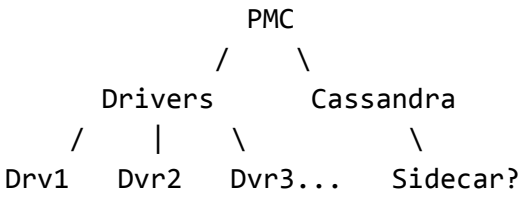
In summary, the subproject approach seems to bring a good trade-off between project independence and coordination, thus appearing as the best option to start with:

- On one side, for most users, drivers are indissociable from the server, and it simply makes more sense to see both hosted together. On a practical level, new feature development will likely require coordination between server and drivers, and future roadmap topics can overlap between server and drivers; by having the drivers as a subproject, CEPs can easily be created for these situations. Similarly, major architectural or API changes in the drivers could impact the server, and thus also require coordination, especially given that some of the donated drivers are being nested and used extensively in the server (internode communication, cqlsh, tests, etc.). By having the drivers in the same project as the server, we can more easily detect and prepare the impacts of such API changes.

- On the other side, by accommodating the drivers in a separate subproject, we still can guarantee a minimal level of independence, especially for daily maintenance and release procedures. See below for in-depth discussion of these matters.

Note that a recurrent concern has been voiced already: current Apache Cassandra committers would have to become knowledgeable of the incoming drivers, and maintain the new code body going forward; this is exacerbated due to different programming languages being incorporated. This legitimate concern will be hopefully mitigated by accepting new driver committers, see "Committership" below.

Also note that it has been considered to further distribute the PMC members in different groups to better differentiate each subproject, e.g.:



However in order to avoid any risk of management overhead, we think that members should fully trust each other to only intervene in domains where they are knowledgeable, and therefore

think that such groups are not necessary, at least for the initial transfer phase. This might of course be reviewed in the future.

B. Source Repositories

Each driver source repository will be transferred to a separate git repository, to be created. Our intention is to donate the entire Git repository of each driver, including all existing commits, branches and tags.

Subprojects are usually hosted in separate source repositories: Apache Spark, Apache Beam and Apache Hadoop for instance have various repositories under the general project umbrella.

The single repo approach was also considered, but we have reasons to believe it will be inappropriate for the present case:

- That was the situation back in 2011 with clearly articulated downsides.
- Release cycles: drivers should keep fairly independent release cycles, which doesn't play nicely with the single repo approach.
- Drivers need to maintain compatibility with a variety of server versions; having drivers and server in the same repo would inevitably lead to constant confusion and overhead about whether a given driver version only works for a given server version (especially if release cycles were coupled, and even more so if versions were aligned, which we do not want – again, see below for in-depth discussion).

Practicalities:

- If possible, we should keep the drivers hosted on GitHub; this way we could grant ownership of the current Github projects to the new GitHub organization, and redirects would be automatically created. This would reduce the disruption for those checking in the drivers codebase, or building them from the source.

C. Commitership

As stated above, we can reasonably assume that the current Apache Cassandra project contributors will not have all the expertise (or bandwidth) to develop drivers in seven different languages.

Several members of the PMC and committers on Cassandra have stated that they think that driver contributors should be made committers on the Cassandra project upon this donation in order to continue developing and maintaining these projects.

Following is an initial list of individuals who have made meaningful contributions to drivers now or in the recent past:

Contributor	Relevant Driver Expertise
Olivier Michallat	Java
Alexandre Dutra	Java
Andrew Tolbert	Java, Node.js
Erik Merkle	Java
Greg Bestland	Java, Python
Tomasz Lelek	Java
Bret McGuire	Java
Adam Holmberg	Python, C++
Alan Boudreault	Python
Jim Witschey	Python
Jorge Bay Gondra	Node.js, C#
Joao Reis	C#
Michael Penick	C++, PHP
Michael Fero	C++, PHP
Sandeep Tamhankar	Ruby, C++, Java
Bulat Shakirzyanov	Ruby, PHP

The same people will need to hold credentials or be assigned owner status of the artifacts in package indices, such as Maven Central, PyPI, NPM and Nuget.

It is worth noting the variety of employers of the above individuals; there is no guarantee that they are still involved on the project nor have a patron to fund their working on the project, and accepting the committer role is a personal decision made on a case-by-case basis.

It is also worth noting that two drivers are currently considered in maintenance mode: PHP and Ruby. This is due mostly to their most active developers not being able to work on these drivers anymore; this situation is unfortunately not expected to change in the near future.

D. Mailing Lists

We suggest that the donated drivers should use the existing Apache Cassandra "user" and "developer" mailing lists, but distinct, per-driver lists for Jira notifications and commits.

We believe that sharing the same mailing list will foster synergies between the drivers and the server; judging from the current usage statistics (more on that below) the overall traffic should stay manageable, but again, we can opt for splitting the mailing lists later if the traffic becomes unmanageable.

The DataStax drivers currently have mailing lists hosted on Google Groups:

- https://groups.google.com/a/lists.datastax.com/g/java-driver-user
- https://groups.google.com/a/lists.datastax.com/g/python-driver-user
- https://groups.google.com/a/lists.datastax.com/g/nodejs-driver-user
- https://groups.google.com/a/lists.datastax.com/g/csharp-driver-user
- https://groups.google.com/a/lists.datastax.com/g/cpp-driver-user
- https://groups.google.com/a/lists.datastax.com/g/php-driver-user
- https://groups.google.com/a/lists.datastax.com/g/ruby-driver-user

Migrating the whole email database seems impractical; we suggest that these groups be closed and users redirected to the "user" mailing list.

E. Issue Tracking

We suggest distinct Jira projects, one per driver, all to be created.

Taking the example of the Sidecar project, the same Jira was used initially but now there is a separate one to track Sidecar specific issues, and we think that it is indeed better to keep Jira issues separated (some drivers may have a rather significant Jira traffic, Java mostly).

The DataStax drivers currently have public Jira projects hosted on datastax-oss.atlassian.net:

- https://datastax-oss.atlassian.net/browse/JAVA
- https://datastax-oss.atlassian.net/browse/PYTHON
- https://datastax-oss.atlassian.net/browse/NODEJS
- https://datastax-oss.atlassian.net/browse/CSHARP
- https://datastax-oss.atlassian.net/browse/CPP
- https://datastax-oss.atlassian.net/browse/PHP
- https://datastax-oss.atlassian.net/browse/RUBY

Migrating the whole Jira database seems intractable; we suggest that these groups be closed and users redirected to the new Jira projects.

F. Documentation

Documentation should move from docs.datastax.com to a new subsection in cassandra.apache.org/doc.

The driver documentation is currently published on the following documentation site: DataStax drivers. It comprises documentation for both OSS and DSE versions of all drivers, as well as documentation for other tools; but in the scope of this CEP, we should consider the following set of documentations only:

- https://docs.datastax.com/en/developer/java-driver/4.7/
- https://docs.datastax.com/en/developer/python-driver/3.24/
- https://docs.datastax.com/en/developer/nodejs-driver/4.5/
- https://docs.datastax.com/en/developer/csharp-driver/3.15/
- https://docs.datastax.com/en/developer/cpp-driver/2.15/
- https://docs.datastax.com/en/developer/php-driver/1.3/
- https://docs.datastax.com/en/developer/ruby-driver/3.2/

Note that the above documentations are all versioned. All versions are considered for donation, and not only the latest. Also note that the above links include generated API documentation as well (e.g. Javadocs).

But there is more to it. The DataStax drivers currently have their doc sources hosted in-tree:

- https://github.com/datastax/java-driver/tree/4.x/manual
- https://github.com/datastax/python-driver/tree/master/docs
- https://github.com/datastax/nodejs-driver/tree/master/doc
- https://github.com/datastax/csharp-driver/tree/master/doc
- https://github.com/datastax/cpp-driver/tree/master/topics
- https://github.com/datastax/php-driver/tree/master/features
- https://github.com/datastax/ruby-driver/tree/master/features

The documentation sources use a variety of doc syntaxes, such as Markdown or reStructuredText. PHP and Ruby use Gherkin feature files. Currently the documentation is authored by the driver committers themselves.

We believe that this in-tree documentation should be kept in place after the transfer to the ASF, and that driver committers should keep owning it.

However, the official drivers documentation hosted on docs.datastax.com is generated from the in-tree, GitHub hosted documentation for each driver, using a proprietary tool called Documentor. It is not in the scope of this CEP to also donate Documentor to the ASF. To complicate things even further, some drivers require manual steps when generating their documentation.

If we want the drivers documentation to be published under cassandra.apache.org/doc, while still keeping the in-tree version, then this will require replacing Documentor and will certainly incur substantial efforts.

Given that the Apache Cassandra documentation is also undergoing major changes in its own CEP, it is unclear at this point how drivers documentation will fit in the new project documentation.

We suggest that:

1. In a first phase (Java driver donation), simply migrate the whole Java driver documentation from docs.datastax.com to cassandra.apache.org/doc.
    a. If a new Java driver release happens in the very near future, we (contributors with access to this proprietary tool) will generate the documentation for the new version using its Documentor tool.
2. When other drivers are transferred, use the same approach.
3. When a solution is found for generating docs from the in-tree sources, apply that solution for all drivers.

Finally, some of the committers will likely need access to the documentation site in order to update the driver docs whenever necessary.

G. Versioning and Release cycle

Drivers will keep an independent release cycle and versioning scheme.

However we expect drivers to integrate in the general project roadmap and to have a release cadence synchronized with that of the server whenever useful; in particular, drivers will strive to release versions supporting new features under development server-side; so that when the feature is released server-side, there already exists at least one supporting driver. A good recent example of that is the work being done to support protocol v5.

Drivers currently use semantic versioning and their numbering is completely decoupled from server versions. This should not change.

Future releases should be proposed, discussed and decided by mail threads on the developer mailing list.

H. Continuous Integration

For the indefinite future, DataStax will continue to test the drivers against Cassandra, DataStax Astra and DataStax Enterprise using existing, private CI infrastructure. Note that DataStax is assessing the viability of making this CI infrastructure public but this is out of the scope of this CEP.

In addition, we will configure jobs that can run on the common CI at ci-cassandra.apache.org, as well as CircleCI.

Practicalities:

- Drivers builds can take up to a few hours when the full integration suite is run against an extensive variety of Cassandra and DSE backends. This is currently done by Jenkins Pipelines multi-job builds.
- Drivers use CCM (Cassandra local cluster manager, written in Python) and Simulacron (Cassandra protocol emulator, written in Java) extensively for their integration tests. The CI containers must have both libraries installed and available on the PATH. It is however not in the scope of this CEP to also donate CCM and/or Simulacron to the ASF.
- Tests related to the DataStax cloud platform Astra also require a predefined Docker image containing a single-node Astra cluster and its proxy.
- Some drivers require building against different platforms, including *nix, Windows, and MacOS.

I. Intellectual Property

As we are not advocating for the drivers to be donated as a separate project, the whole incubation procedure is not required.

But this does not mean that the donated code is ready to be integrated. Instead, we will have to abide by the "lightweight" incubation procedure described here, which is "designed to allow code to be imported with alacrity while still providing for oversight".

As the clearance document states, "the receiving PMC is responsible for doing the work".

New or Changed Public Interfaces

The existing Cassandra codebase should see no immediate changes, if the guidelines below for compatibility and migration are fully executed.

Compatibility, Deprecation, and Migration Plan

A. Driver APIs

In order to minimize the disruption caused to users by the donation, we suggest that all the following items remain unchanged:

- Public API: class and interface names, package names and namespaces.
- Distribution: artifact names as they appear in Maven Central, PyPI, NPM, Nuget, etc.

Note that, while most drivers usually declare their API for Apache Cassandra inside a `cassandra` namespace or equivalent, there are exceptions:

- Java driver 4.x declares its API for Apache Cassandra under the root package `com.datastax.oss.driver` (note the `datastax` word).
- Java driver 3.x declares its API for Apache Cassandra under the root package `com.datastax.driver` (again, note the `datastax` word).

Other drivers usually reserve the word `datastax` for proprietary APIs (see below).

Similarly, most artifact names only contain the word `cassandra`, e.g. the Python driver. And again, the Java driver is an exception:

- Java driver 3 has group ID `com.datastax.cassandra` and artifact ID `cassandra-driver-core`.
- Java driver 4 has group ID `com.datastax.oss` and artifact ID `java-driver-core`.

We suggest that the word `datastax` be left as is both in the Java driver API and in artifact names in order to avoid disruption until a future major revision release. We have confirmed that this is viable in an inquiry to the Apache Legal group.

B. DataStax proprietary software

For a long time, DataStax maintained, for each programming language, two different driver flavors: one for Apache Cassandra, and another for DataStax Enterprise (DSE).

This has been identified as a source of confusion for users. In January 2020, DataStax published the first version of its drivers where DSE-specific functionality was merged into the OSS drivers, resulting in what is commonly called "the unified drivers".

Unified drivers are thus capable of connecting to any CQL-compatible backend: Apache Cassandra of course, but also DSE and Astra, and include DataStax proprietary technologies such as DSE authentication mechanisms, DSE continuous paging, and DSE Graph. These drivers are also able to connect using DSE-specific versions of the CQL native protocol.

DSE-specific features are usually clearly demarcated in the drivers APIs behind specific namespaces or packages, usually containing the word `datastax` or `dse`.

We suggest that these features be donated as well. Indeed, the removal of such features, a few months only after their inclusion in the unified drivers, would be a source of confusion for users. We as driver contributors are committed to not including any more proprietary features in the drivers once they are transferred to the ASF.

Note that some of the DataStax-specific features require external dependencies; and notably DSE Graph, which requires Apache Tinkerpop artifacts. In the case of the Java driver, this is a series of Maven artifacts that the driver declares as mandatory dependencies; however, it is smart enough to live without them if they are excluded. In the case of the other drivers, this is a dependency on the corresponding GLV, and can be excluded as well.

C. Apache Cassandra internal usage of the drivers

Apache Cassandra itself uses two drivers internally: Java 3.x and Python 3.x.

As we mentioned already, Java 3.x is currently in maintenance mode. We as driver developers are offering to support the 3.x series until the whole server codebase is migrated to Java version 4, and continue to help during the migration period of users and the project to the new Java Driver version. For context on why we strongly advocate for the use of the new driver, this blog post on the topic was published when Java driver 4 was released in 2019.

The Python driver is currently working on its new major version, 4.0. We are similarly offering to migrate the server codebase when the driver release is ready.

Test Plan

Describe in few sentences how the CEP will be tested. We are mostly interested in system tests (since unit-tests are specific to implementation details). How will we know that the implementation works as expected? How will we know nothing broke?

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.