antlr / antlr4 Public

Code | Issues 616 | Pull requests 75 | Discussions | Actions | Project

Releases | Tags

This is a draft and won’t be seen by the public unless it’s published. Discard draft

4.10-rc.1

Target: master

Excellent! This tag will be created from the target when you publish this release.

4.10 Major feature, code clean up, and bug fix release

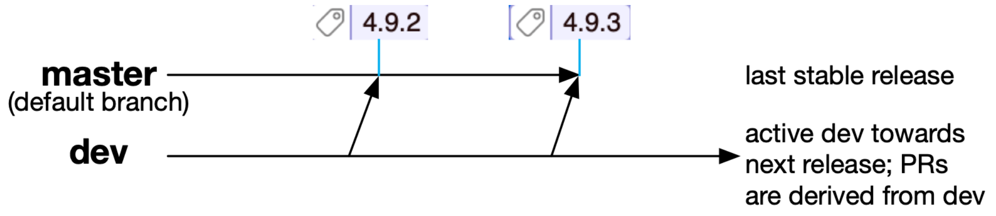
Write | Preview

There were no pull requests associated with the commits included in this release. Automatically-generated notes were not generated.

This is a very major release with a number of important changes. Grammars generated by ANTLR tool 4.10 will not work with runtime libraries prior to 4.10. Please regenerate your parsers and use the latest runtime library for your target. There have been many valuable contributions, but I welcome **@KvanTTT** and **@jcking** as recent "official" major ANTLR contributors. :)

Repo branching structure

We have changed the branching structure of the repository. The default branch for this repo remains `master` and it is the latest stable release with tags for the various releases; e.g., see release tag 4.9.3. We now do development work in branch `dev` between releases and all pull requests should be derived from that branch. The `dev` branch is merged back into `master` to cut a release and the release state is tagged (e.g., with `4.10-rc.1` or `4.10`.) Visually our process looks roughly like this:



Targets such as Go that pull directly from the repository can use the default `master` branch but can also pull from the active `dev` branch:

```
$ go get github.com/antlr/antlr4/runtime/Go/antlr@dev
```

Developers certificate of origin

In order to bring ANTLR more in line with current standard standards for contribution processes, as of 4.10, ANTLR uses the Linux Foundation's Developer Certificate of Origin, DCO, version 1.1. See file https://github.com/antlr/antlr4/raw/master/developer-cert-of-origin.txt . It is simpler than the original contributors license agreement, which required programmers to sign the `contributors.txt` file, which has now moved to file historical-contributors-agreement.txt .

Each commit in pull requests must have a "signature", which is simple as using `-s` (not `-S`) on the git commit command:

```
$ git commit -s -m 'This is my commit message'
```

Github's pull request process enforces the sig and gives instructions on how to fix any commits that lack the sig. See https://github.com/apps/dco for more info.

4.10-generated parsers incompatible with previous runtimes

ANTLR not only generates recursive-descent parsers; it generates a state machine called an augmented transition network (ATN) in serialized form as a bunch of integers stored in the generated parser and lexer files. This serialization format was changed for 4.10 to remove a size limit on the supported ATNs. See #3591.

The key point here is that we changed the version number stored inside the serialization format and so, in order to use this new version of ANTLR, you must regenerate all of your lexers and parsers using the 4.10 tool and then use the new runtime. Parsers generated with 4.10 or not compatible with previous versions of the runtime.

Increasing minimum java version

Going forward, we are using Java 11 for the source code and the compiled .class files for the ANTLR tool. The Java runtime target, however, and the associated runtime tests use Java 8 (bumping up from Java 7).

Miscellaneous improvements

- rule options; case-insensitive
- improved error messages
- we rebuilt the runtime testing rig and made continuous integration improvements; part of this was done to support more recent versions of Java
- we cleaned up the maven pom and brought it up to date (still has errors if somebody can make a suggestion.)

Issues fixed

- do we need this C++ ATN serialization? (target:cpp, type:cleanup)
- Incorrect type of token with number 0xFFFF because of incorrect ATN serialization (atn-analysis, type:bug)
- Clean up ATN serialization: rm UUID and shifting by value of 2 (atn-analysis, type:cleanup)
- The parseFile method of the InterpreterDataReader class is missing code: "line = br.readLine();" (type:bug, target:java)
- Serialized ATN data element 810567 element 11 out of range 0..65535 (atn-analysis, type:cleanup)
- Go target, unable to check when custom error strategy is in recovery mode (target:go)
- Escape issue for characeters (grammars, type:bug)
- UnsupportedOperationException while generating code for large grammars. (atn-analysis, type:cleanup)
- Add a more understandable message than "Serialized ATN data element .... element ... out of range 0..65535" (atn-analysis, type:cleanup)
- avoid java.lang.StackOverflowError (lexers, error-handling)
- Getting this error: Exception in thread "main" java.lang.UnsupportedOperationException: Serialized ATN data element out of range (atn-analysis, type:cleanup)

Improvements, features

- Escape bad words during grammar generation (code-gen, type:improvement)
- Implement caseInsensitive option (lexers, options, type:improvement)
- Some tool bugfixes (error-handling, comp:tool, type:improvement, type:cleanup)

Pull requests grouped by target

csharp target

- use ints not chars for C# (code-gen, atn-analysis, target:csharp)
- C# and Go runtime tests fixes on non-Windows OS (comp:runtime, target:csharp, target:go, comp:testing)

- C#, JavaScript, Go runtime tests performance improvements (target:javascript, target:csharp, target:go, comp:performance)

cpp target

- [C++] Fix bugs in SemanticContext (target:cpp, comp:performance)
- [C++] Add ANTLR4CPP_PUBLIC attributes to various symbols (target:cpp, type:cleanup)
- Update editorconfig for c++ (target:cpp)
- [C++] Avoid using dynamic_cast where possible by using hand rolled RTTI (actions, target:cpp, type:cleanup)
- [C++] Fix generated Lexer static data constructor (target:cpp, type:cleanup)
- [C++] Force generated static data type name to titlecase (target:cpp, type:cleanup)
- [C++] Remove unused LexerATNSimulator::match_calls (target:cpp, type:cleanup)
- [C++] Switch to camel case for generated static data variable names (target:cpp, type:cleanup)
- [C++] Remove duplicate includes and remove unused includes (target:cpp, type:cleanup)
- [C++] Lazily deserialize ATN in generated code (target:cpp, type:cleanup)
- [C++] Cleanup ATNDeserializer and remove related deprecated methods from ATNSimulator (target:cpp, type:cleanup)
- [C++] Improve multithreaded performance, fix TSAN error, and fix profiling ATN simulator setup (target:cpp)
- [C++] Remove now unused antlrcpp::Guid (target:cpp)
- [C++] Fix const correctness in ATN and DFA (target:cpp)
- [C++] Fix missing virtual destructors (target:cpp, type:cleanup)
- [C++] Optimize LL1Analyzer (target:cpp)
- [C++] Update cmake README.md to C++17 (comp:build, target:cpp)
- [C++] Fix Windows min/max macro collision (target:cpp)
- [C++] Optimizations and cleanups and const correctness, oh my (target:cpp, type:cleanup)
- cpp/cmake: Fix library install directories (target:cpp)
- [C++] Improve MurmurHash performance for Clang (target:cpp, comp:performance)
- [C++] Fix bugs in UnbufferedCharStream (target:cpp)
- [C++] Improve whitespace escaping (target:cpp, type:cleanup)
- [C++] Clean up antlr4::misc::Interval and antlr4::dfa (target:cpp)
- Fixed C++ runtime demo (target:cpp)
- [C++] Optimize ATN deserialization (target:cpp, comp:performance)
- [C++] Cleanup EMPTY_VOCABULARY uses and remove SingleWriteMultipleRea… (target:cpp)
- [C++] Implement standalone Unicode encoding and decoding handling (target:cpp)
- [C++] Switch to std::any and deprecate antlrcpp::Any (target:cpp)
- [C++] Fix ambiguous ANTLRInputStream constructor overload (target:cpp)
- [C++] Partially undo commit a8d828016e54adfc403326d8e56d93e7c75c27fc (target:cpp)
- [C++] Modernize Vocabulary using std::string_view (target:cpp)
- [C++] Switch from NOEXCEPT to noexcept (target:cpp)

- Fix reported memory leak (target:cpp)
- Add .clang-format file to ensure consistent style of C++ (target:cpp)
- [C++] Bump minimum required version to C++17 (target:cpp)
- [C++] Upstream various performance, build, and best practice fixes (target:cpp)
- [C++] Cleanup ATNDeserializer interface (target:cpp)
- Fix TSAN warnings related to dfa::DFA::s0 (target:cpp)
- Corrected/clarified the Linux C++ runtime compilation instructions (comp:build, comp:doc, target:cpp)

go target

- [Go] Do not export Array2DHashSet which is an implementation detail (target:go, type:cleanup)
- [Go] Cleanup and fix ATN deserialization verification (target:go, type:cleanup)
- Don't decode utf16 in ATN deserialization for Go; it's just a bunch of uint16 (atn-analysis, target:go)
- [Go] Port locking algorithm from C++ to Go (target:go, threading)
- perf: Add default implementation for Visit in ParseTreeVisitor. (trees-contexts, target:go)
- [Go] Fix for issue 3557 -- get "go test" working again (target:go, type:cleanup)
- make InErrorRecoveryMode public for go (target:go)
- Fix bail error strategy nil interface panic (trees-contexts, type:bug, target:go)
- [Go] Fix (*BitSet).equals (target:go)
- [Go] Fix BitSet OR operation (target:go)
- [Go] Make BaseGoTest::initializeRuntime static (target:go, comp:testing)
- [Go] Improve BitSet implementation (target:go)
- C# and Go runtime tests fixes on non-Windows OS (comp:runtime, target:csharp, target:go, comp:testing)
- C#, JavaScript, Go runtime tests performance improvements (target:javascript, target:csharp, target:go, comp:performance)
- Get rid of reflection in Go.stg (target:go)
- Use atomic instead of a shared mutex for DFA.s0 (target:go)

java target

- [Java] Preserve serialized ATN version 3 compatibility (atn-analysis, target:java)
- Fix .interp file parsing test for the Java runtime. (target:java)
- Remove unused variable from generated code (target:java)
- Update javadoc to 8 and 11 (comp:build, target:java, comp:doc)
- use src 11 for tool, but 8 for plugin/runtime (comp:build, target:java)
- [Java] Fix remaining complaints from ErrorProne static analysis (target:java, type:cleanup)

javascript target

- JavaScript: Change string to int array for serialized ATN (code-gen, atn-analysis, target:javascript, type:cleanup)

- [JavaScript] Migrate from jest to jasmine (target:javascript)
- Add browser fs fallback (target:javascript)
- C#, JavaScript, Go runtime tests performance improvements (target:javascript, target:csharp, target:go, comp:performance)

python2 target

- Convert Python2/3 to use int arrays not strings for ATN encodings (atn-analysis, target:python2, target:python3)

python3 target

- Convert Python2/3 to use int arrays not strings for ATN encodings (atn-analysis, target:python2, target:python3)
- [Python3] rename pygrun, add entry_point (comp:build, target:python3)
- Python local import fixes (comp:build, target:python3)

swift target

- [Docs] Update Swift Docs (comp:doc, target:swift)
- Use release build config instead of debug for Swift runtime tests (improve performance more than 50%) (target:swift, comp:testing, comp:performance)
- Update Swift template to silence warnings about unmodified class reference (target:swift)
- Update Swift runtime doc (comp:doc, target:swift)
- Use `AnyObject` instead of `class`. (target:swift)
- [Swift] Use proper constants (target:swift)
- Add Support For Swift Package Manager (comp:build, target:swift)

php target

- Fix code style and typing to support PHP 8 (target:php, type:cleanup)
- Fix PHP codegen to support int ATN serialization (atn-analysis, target:php, type:cleanup)

dart target

- Dart int atn (atn-analysis, target:dart, type:cleanup)
- Fix Dart linter errors under strict mode (target:dart, type:cleanup)
- Remove `late` modifier for Dart generated context classes (target:dart)

Contributors

- 50 items: Justin King (jcking)
- 26 items: Terence Parr (parrt)

- 15 items: Ivan Kochurkin (KvanTTT)
- 2 items: Marcos Passos (marcospassos)
- 2 items: ericvergnaud
- 2 items: 100mango
- 2 items: Tinker (Tinker1024)
- 1 items: Henner Zeller (hzeller)
- 1 items: chen quan (chenquan)
- 1 items: Ken Domino (kaby76)
- 1 items: Matt Willer (mattwiller)
- 1 items: Biswapriyo Nath (Biswa96)
- 1 items: Alexandr (Alex-Andrv)
- 1 items: Oleksandr Martyshchenko (Ketler13)
- 1 items: Mike Lischke (mike-lischke)
- 1 items: Eneko Alonso (eneko)
- 1 items: Nicholas Bollweg (bollwyvl)
- 1 items: hlstwizard
- 1 items: Yoshimasa Niwa (niw)
- 1 items: frog酋长 (ccrfrog)
- 1 items: Dante Broggi (Dante-Broggi)
- 1 items: Isaac (idraper)
- 1 items: Roman Gafurov (seth2810)
- 1 items: Benjamin Spiegel (SkittlePox)
- 1 items: Abe149
- 1 items: Karl Heinz Marbaise (khmarbaise)
- 1 items: Roger Far (rogerfar)
- 1 items: Miguel Alejandro Muñoz Gil (mimuogi)
- 1 items: Harry Chan (re-thc)
- 1 items: Jérôme BAROTIN (jbarotin)
- 1 items: Wolfgang Fahl (WolfgangFahl)
- 1 items: Andrea Giovacchini (agiovacchini)

Full Changelog: `4.9.3...4.10`

↓ Attach binaries by dropping them here or selecting them.

☑ **This is a pre-release**
We'll point out that this release is identified as non-production ready.

☐ **Create a discussion for this release**
People will be able to leave comments and reactions on this release using Discussions.

Publish release Save draft

Tagging suggestions

It’s common practice to prefix your version names with the letter `v`. Some good tag names might be `v1.0.0` or `v2.3.4`.

If the tag isn’t meant for production use, add a pre-release version after the version name. Some good pre-release versions might be `v0.2.0-alpha` or `v5.9-beta.3`.

Semantic versioning

If you’re new to releasing software, we highly recommend reading about semantic versioning.