

1.14.0

Features

- Defer running top-level container nodes until RunSpecs is called [d44dedf]
- [Document Ginkgo lifecycle](#)
- Add `extensions/globals` package (#692) [3295c8f] - this can be helpful in contexts where you are test-driving your test-generation code (see [#692](#))
- Print Skip reason in JUnit reporter if one was provided [820dfab]

1.13.0

Features

- Add a version of `table.Entry` that allows dumping the entry parameters. (#689) [21eaef2]

Fixes

- Ensure integration tests pass in an environment sans GOPATH [606fba2]
- Add books package (#568) [fc0e44e]
- `doc(readme)`: installation via "tools package" (#677) [83bb20e]
- Solve the undefined: `unix.Dup2` compile error on mips64le (#680) [0624f75]
- Import package without dot (#687) [6321024]
- Fix integration tests to stop require GOPATH (#686) [a912ec5]

1.12.3

Fixes

- Print correct code location of failing table test (#666) [c6d7afb]

1.12.2

Fixes

- Update dependencies [ea4a036]

1.12.1

Fixes

- Make unfocus ("blur") much faster (#674) [8b18061]
- Fix typo (#673) [7fdcbe8]
- Test against 1.14 and remove 1.12 [d5c2ad6]
- Test if a coverprofile content is empty before checking its latest character (#670) [14d9fa2]
- replace tail package with maintained one. this fixes go get errors (#667) [4ba33d4]
- improve ginkgo performance - makes progress on #644 [a14f98e]
- fix convert integration tests [1f8ba69]
- fix typo succesful -> successful (#663) [1ea49cf]
- Fix invalid link (#658) [b886136]
- convert utility : Include comments from source (#657) [1077c6d]
- Explain what BDD means [d79e7fb]
- skip race detector test on unsupported platform (#642) [f8ab89d]
- Use `Dup2` from `golang.org/x/sys/unix` instead of `syscall.Dup` (#638) [5d53c55]
- Fix missing newline in combined coverage file (#641) [6a07ea2]

- check if a spec is run before returning SpecSummary (#645) [8850000]

1.12.0

Features

- Add module definition (#630) [78916ab]

1.11.0

Features

- Add syscall for riscv64 architecture [f66e896]
- teamcity reporter: output location of test failure as well as test definition (#626) [9869142]
- teamcity reporter: output newline after every service message (#625) [3cfa02d]
- Add support for go module when running `generate` command (#578) [9c89e3f]

1.10.3

Fixes

- Set `go_import_path` in `travis.yml` to allow internal packages in forks (#607) [3b721db]
- Add integration test [d90e0dc]
- Fix coverage files combining [e5dde8c]
- A new CLI option: `-ginkgo.reportFile` (#601) [034fd25]

1.10.2

Fixes

- speed up table entry `generatelt()` (#609) [5049dc5]
- Fix. Write errors to `stderr` instead of `stdout` (#610) [7bb3091]

1.10.1

Fixes

- stack backtrace: fix skipping (#600) [2a4c0bd]

1.10.0

Fixes

- stack backtrace: fix alignment and skipping [66915d6]
- fix typo in documentation [8f97b93]

1.9.0

Features

- Option to print output into report, when tests have passed [0545415]

Fixes

- Fixed typos in comments [0ecbc58]
- gofmt code [a7f8bfb]
- Simplify code [7454d00]

- Simplify concatenation, incrementation and function assignment [4825557]
- Avoid unnecessary conversions [9d9403c]
- JUnit: include more detailed information about panic [19cca4b]
- Print help to stdout when the user asks for help [4cb7441]

1.8.0

New Features

- allow config of the vet flag for `go test` (#562) [3cd45fa]
- Support projects using go modules [d56ee76]

Fixes and Minor Improvements

- chore(godoc): fixes typos in Measurement funcs [dbaca8e]
- Optimize focus to avoid allocations [f493786]
- Ensure generated test file names are underscored [505cc35]

1.7.0

New Features

- Add JustAfterEach (#484) [0d4f080]

Fixes

- Correctly round suite time in junit reporter [2445fc1]
- Avoid using -i argument to go test for Golang 1.10+ [46bbc26]

1.6.0

New Features

- add --debug flag to emit node output to files (#499) [39febac]

Fixes

- fix: for `go vet` to pass [69338ec]
- docs: fix for contributing instructions [7004cb1]
- consolidate and streamline contribution docs (#494) [d848015]
- Make generated Junit file compatible with "Maven Surefire" (#488) [e51bee6]
- all: gofmt [000d317]
- Increase eventually timeout to 30s [c73579c]
- Clarify asynchronous test behaviour [294d8f4]
- Travis badge should only show master [26d2143]

1.5.0 5/10/2018

New Features

- Supports go v1.10 (#443, #446, #451) [e873237, 468e89e, e37dbfe, a37f4c0, c0b857d, bca5260, 4177ca8]
- Add a When() synonym for Context() (#386) [747514b, 7484dad, 7354a07, dd826c8]
- Re-add noisySkippings flag [652e15c]
- Allow coverage to be displayed for focused specs (#367) [11459a8]
- Handle -outputdir flag (#364) [228e3a8]
- Handle -coverprofile flag (#355) [43392d5]

Fixes

- When using custom reporters register the custom reporters *before* the default reporter. This allows users to see the output of any print statements in their custom reporters. (#365) [8382b23]
- When running a test and calculating the coverage using the `-coverprofile` and `-outputdir` flags, Ginkgo fails with an error if the directory does not exist. This is due to an [issue in go 1.10](#) (#446) [b36a6e0]
- `unfocus` command ignores vendor folder (#459) [e5e551c, c556e43, a3b6351, 9a820dd]
- Ignore packages whose tests are all ignored by go (#456) [7430ca7, 6d8be98]
- Increase the threshold when checking time measurements (#455) [2f714bf, 68f622c]
- Fix race condition in coverage tests (#423) [a5a8ff7, ab9c08b]
- Add an extra new line after reporting spec run completion for test2json [874520d]
- added name field to junit reported testsuite [ae61c63]
- Do not set the run time of a spec when the dryRun flag is used (#438) [457e2d9, ba8e856]
- Process FWhen and FSpecify when unfocusing (#434) [9008c7b, ee65bd, df87dfe]
- Synchronise the access to the state of specs to avoid race conditions (#430) [7d481bc, ae6829d]
- Added Duration on GinkgoTestDescription (#383) [5f49dad, 528417e, 0747408, 329d7ed]
- Fix Ginkgo stack trace on failure for Specify (#415) [b977ede, 65ca40e, 6c46eb8]
- Update README with Go 1.6+, Golang -> Go (#409) [17f6b97, bc14b66, 20d1598]
- Use `fmt.Errorf` instead of `errors.New(fmt.Sprintf` (#401) [a299f56, 44e2eaa]
- Imports in generated code should follow conventions (#398) [0bec0b0, e8536d8]
- Prevent data race error when Recording a benchmark value from multiple go routines (#390) [c0c4881, 7a241e9]
- Replace GOPATH in Environment [4b883f0]

1.4.0 7/16/2017

- `ginkgo` now provides a hint if you accidentally forget to run `ginkgo bootstrap` to generate a `*_suite_test.go` file that actually invokes the Ginkgo test runner. [#345](#)
- thanks to improvements in `go test -c ginkgo` no longer needs to fix Go's compilation output to ensure compilation errors are expressed relative to the CWD. [#357](#)
- `ginkgo watch -watchRegExp=...` allows you to specify a custom regular expression to watch. Only files matching the regular expression are watched for changes (the default is `\.go$`) [#356](#)
- `ginkgo` now always emits compilation output. Previously, only failed compilation output was printed out. [#277](#)
- `ginkgo -requireSuite` now fails the test run if there are `*_test.go` files but `go test` fails to detect any tests. Typically this means you forgot to run `ginkgo bootstrap` to generate a suite file. [#344](#)
- `ginkgo -timeout=DURATION` allows you to adjust the timeout for the entire test suite (default is 24 hours) [#248](#)

1.3.0 3/28/2017

Improvements:

- Significantly improved parallel test distribution. Now instead of pre-sharding test cases across workers (which can result in idle workers and poor test performance) Ginkgo uses a shared queue to keep all workers busy until all tests are complete. This improves test-time performance and consistency.
- `Skip(message)` can be used to skip the current test.
- Added `extensions/table` - a Ginkgo DSL for [Table Driven Tests](#)
- Add `GinkgoRandomSeed()` - shorthand for `config.GinkgoConfig.RandomSeed`
- Support for retrying flaky tests with `--flakeAttempts`

- `ginkgo ./...` now recurses as you'd expect
- Added `Specify` a synonym for `It`
- Support colorise on Windows
- Broader support for various go compilation flags in the `ginkgo` CLI

Bug Fixes:

- Ginkgo tests now fail when you `panic(nil)` (#167)

1.2.0 5/31/2015

Improvements

- `ginkgo -coverpkg` calls down to `go test -coverpkg` (#160)
- `ginkgo -afterSuiteHook COMMAND` invokes the passed-in `COMMAND` after a test suite completes (#152)
- Relaxed requirement for Go 1.4+. `ginkgo` now works with Go v1.3+ (#166)

1.2.0-beta

Ginkgo now requires Go 1.4+

Improvements:

- Call reporters in reverse order when announcing spec completion -- allows custom reporters to emit output before the default reporter does.
- Improved focus behavior. Now, this:

```
FDescribe("Some describe", func() {
    It("A", func() {})

    Fit("B", func() {})
})
```

will run `B` but *not* `A`. This tends to be a common usage pattern when in the thick of writing and debugging tests.

- When `SIGINT` is received, Ginkgo will emit the contents of the `GinkgoWriter` before running the `AfterSuite`. Useful for debugging stuck tests.
- When `--progress` is set, Ginkgo will write test progress (in particular, Ginkgo will say when it is about to run a `BeforeEach`, `AfterEach`, `It`, etc...) to the `GinkgoWriter`. This is useful for debugging stuck tests and tests that generate many logs.
- Improved output when an error occurs in a setup or teardown block.
- When `--dryRun` is set, Ginkgo will walk the spec tree and emit to its reporter *without* actually running anything. Best paired with `-v` to understand which specs will run in which order.
- Add `By` to help document long `It` s. `By` simply writes to the `GinkgoWriter`.
- Add support for precompiled tests:

- `ginkgo build <path-to-package>` will now compile the package, producing a file named `package.test`
- The compiled `package.test` file can be run directly. This runs the tests in series.
- To run precompiled tests in parallel, you can run: `ginkgo -p package.test`
- Support `bootstrap` ping and `generate` ing [Agouti](#) specs.
- `ginkgo generate` and `ginkgo bootstrap` now honor the package name already defined in a given directory
- The `ginkgo` CLI ignores `SIGQUIT` . Prevents its stack dump from interlacing with the underlying test suite's stack dump.
- The `ginkgo` CLI now compiles tests into a temporary directory instead of the package directory. This necessitates upgrading to Go v1.4+.
- `ginkgo -notify` now works on Linux

Bug Fixes:

- If `--skipPackages` is used and all packages are skipped, Ginkgo should exit 0.
- Fix tempfile leak when running in parallel
- Fix incorrect failure message when a panic occurs during a parallel test run
- Fixed an issue where a pending test within a focused context (or a focused test within a pending context) would skip all other tests.
- Be more consistent about handling `SIGTERM` as well as `SIGINT`
- When interrupted while concurrently compiling test suites in the background, Ginkgo now cleans up the compiled artifacts.
- Fixed a long standing bug where `ginkgo -p` would hang if a process spawned by one of the Ginkgo parallel nodes does not exit. (Hooray!)

1.1.0 (8/2/2014)

No changes, just dropping the beta.

1.1.0-beta (7/22/2014)

New Features:

- `ginkgo watch` now monitors packages *and their dependencies* for changes. The depth of the dependency tree can be modified with the `-depth` flag.
- Test suites with a programmatic focus (`Fit` , `FDescribe` , etc...) exit with non-zero status code, even when they pass. This allows CI systems to detect accidental commits of focused test suites.
- `ginkgo -p` runs the testsuite in parallel with an auto-detected number of nodes.
- `ginkgo -tags=TAG_LIST` passes a list of tags down to the `go build` command.
- `ginkgo --failFast` aborts the test suite after the first failure.
- `ginkgo generate file_1 file_2` can take multiple file arguments.
- Ginkgo now summarizes any spec failures that occurred at the end of the test run.
- `ginkgo --randomizeSuites` will run tests *suites* in random order using the generated/passed-in seed.

Improvements:

- `ginkgo -skipPackage` now takes a comma-separated list of strings. If the *relative path* to a package matches one of the entries in the comma-separated list, that package is skipped.
- `ginkgo --untilItFails` no longer recompiles between attempts.
- Ginkgo now panics when a runnable node (`It` , `BeforeEach` , `JustBeforeEach` , `AfterEach` , `Measure`) is nested within another runnable node. This is always a mistake. Any test suites that panic because of this change should be fixed.

Bug Fixes:

- `ginkgo bootstrap` and `ginkgo generate` no longer fail when dealing with hyphen-separated packages .
- parallel specs are now better distributed across nodes - fixed a crashing bug where (for example) distributing 11 tests across 7 nodes would panic

1.0.0 (5/24/2014)

New Features:

- Add `GinkgoParallelNode()` - shorthand for `config.GinkgoConfig.ParallelNode`

Improvements:

- When compilation fails, the compilation output is rewritten to present a correct *relative* path. Allows ⌘-clicking in iTerm open the file in your text editor.
- `--untilItFails` and `ginkgo watch` now generate new random seeds between test runs, unless a particular random seed is specified.

Bug Fixes:

- `-cover` now generates a correctly combined coverprofile when running with in parallel with multiple `-node` s.
- Print out the contents of the `GinkgoWriter` when `BeforeSuite` or `AfterSuite` fail.
- Fix all remaining race conditions in Ginkgo's test suite.

1.0.0-beta (4/14/2014)

Breaking changes:

- `thirdparty/gomocktestreporter` is gone. Use `GinkgoT()` instead
- Modified the Reporter interface
- `watch` is now a subcommand, not a flag.

DSL changes:

- `BeforeSuite` and `AfterSuite` for setting up and tearing down test suites.
- `AfterSuite` is triggered on interrupt (`^C`) as well as exit.
- `SynchronizedBeforeSuite` and `SynchronizedAfterSuite` for setting up and tearing down singleton resources across parallel nodes.

CLI changes:

- `watch` is now a subcommand, not a flag
- `--nodot` flag can be passed to `ginkgo generate` and `ginkgo bootstrap` to avoid dot imports. This explicitly imports all exported identifiers in Ginkgo and Gomega. Refreshing this list can be done by running `ginkgo nodot`

- Additional arguments can be passed to specs. Pass them after the `--` separator
- `--skipPackage` flag takes a regexp and ignores any packages with package names passing said regexp.
- `--trace` flag prints out full stack traces when errors occur, not just the line at which the error occurs.

Misc:

- Start using semantic versioning
- Start maintaining changelog

Major refactor:

- Pull out Ginkgo's internal to `internal`
- Rename `example` everywhere to `spec`
- Much more!