Custom clang-tidy checks, from YAML to CI

```
CONTRIBUTING.md:42:1: warning: style is only 'mostly' automatically checked [misc-eyeball-diff]
```

Aleksei Markelov · Software Engineer at TomTom

Your guide's rule may already be a check

built-in → query → plugin

406 checks ship with clang-tidy, **219** with fixes.

.clang-tidy

```
Checks: 'readability-identifier-naming'
CheckOptions:
  readability-identifier-naming.PrivateMemberCase: camelBack
  readability-identifier-naming.PrivateMemberSuffix: '_'
```

include/cache.hpp

```
class Cache {
  int capacity;
};
```

```
cache.hpp:2:7: warning: invalid case style for private member
'capacity' [readability-identifier-naming]
    2 |   int capacity;
      |       ^~~~~~~~
      |       capacity_
```

Write it in clang-query, paste it into YAML

built-in → **query** → plugin CustomChecks · clang-tidy 22+ · experimental

include/telemetry/sample.hpp

```
struct Sample {
  double speed;        double heading_deg;
  float  altitude;     int    satellites;
};
```

```
clang-query> match fieldDecl(
    isExpansionInFileMatching("/include/"))
                                    4 matches
clang-query> match fieldDecl(…,
    hasType(realFloatingPointType()))
                                    3 matches
clang-query> match fieldDecl(…,
    unless(matchesName("(_m|_mps|_deg|_s|_ms)$")))
  speed  altitude                   2 matches
```

.clang-tidy

```
Checks: 'custom-unit-suffix'
CustomChecks:
  - Name: unit-suffix
    Query: |
      match fieldDecl(
        isExpansionInFileMatching("/include/"),
        unless(isImplicit()),
        hasType(hasUnqualifiedDesugaredType(
          realFloatingPointType())),
        unless(matchesName("(_m|_mps|_deg|_s|_ms)$"))
      ).bind("field")
    Diagnostic:
      - BindName: field
        Message: "floating-point field in a public
                  header needs a unit suffix"
        Level: Warning
```

Where a strong type becomes a raw number

built-in → query → **plugin** plugin · clang-tidy 14+ · -load

```
struct RetryPolicy { long timeout_ms; };

void Configure(RetryPolicy& policy,
               std::chrono::seconds timeout) {
  policy.timeout_ms = timeout.count();
}
```

```
boundary.cpp:8:3: warning: 'timeout_ms' expects milliseconds
but receives a raw value in seconds [units-boundary]
    8 |   policy.timeout_ms = timeout.count();
      |   ^
```

Why a plugin: it does arithmetic on `std::ratio`, and its message names both units. A query can only list every pair and print one fixed text.

Start with a query. Reach for a plugin when the rule needs logic or more than one repo. Upstream it when everyone needs it.

Warn first, block later, one directory at a time

A new rule doesn't need a rewrite

1

Warn in one directory

`src/telemetry/.clang-tidy`

```
InheritParentConfig: true
Checks: 'custom-unit-suffix'
CustomChecks: …
```

2

Clean it up, then block there

`same file`

```
InheritParentConfig: true
Checks: 'custom-unit-suffix'
WarningsAsErrors: 'custom-unit-
suffix'
```

3

Warn repo-wide

`root .clang-tidy`

```
Checks: '…,custom-unit-suffix'
CustomChecks: …
```

tomtom

Pick one rule your guide has, and you have seen broken last week.

Give your reviewers better problems to solve.