



libgit2

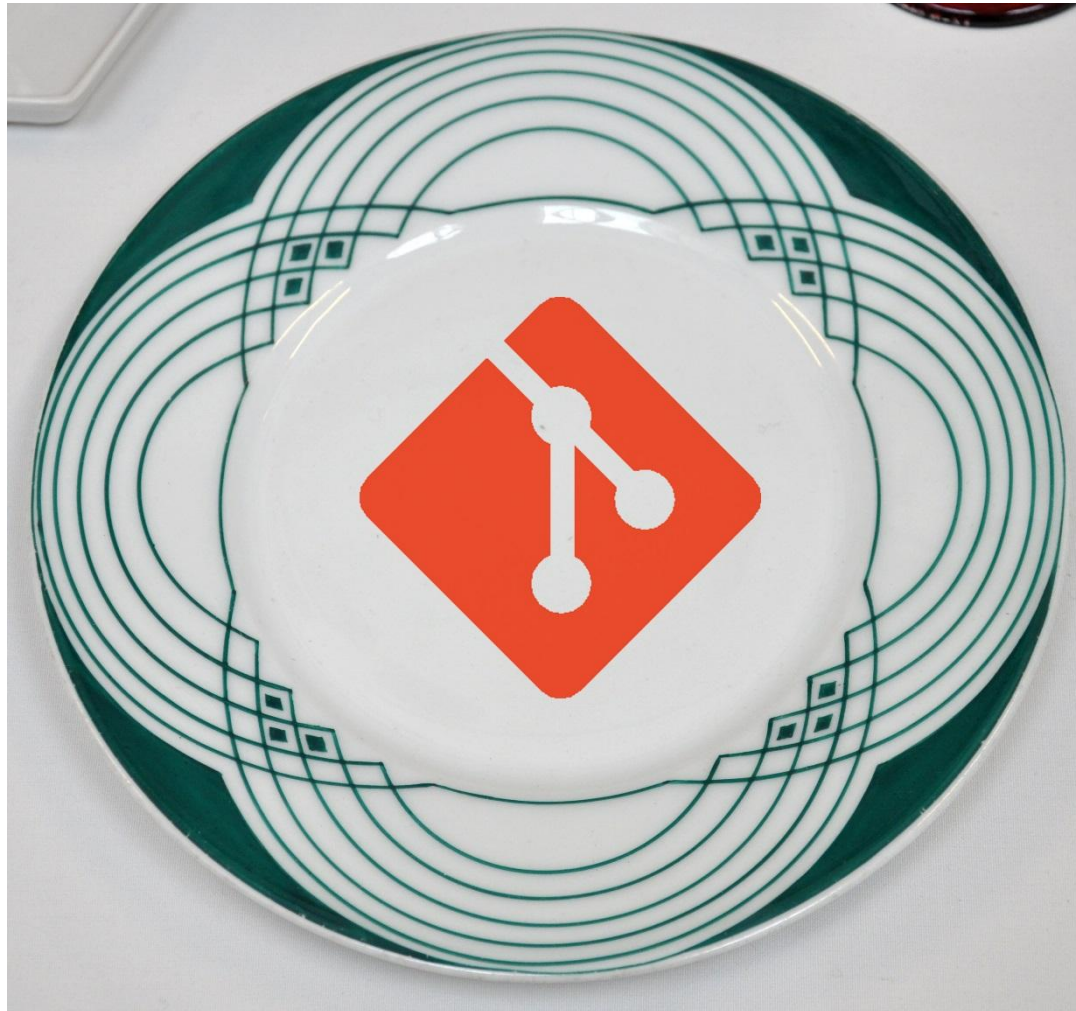
Carlos Martín Nieto

elego Software Solutions GmbH - © 2013

- Carlos Martín
- Git & libgit2 since 2011
- elego Software Solutions GmbH



```
% ls -l ~/libexec/git-core | wc -l  
169
```





```
% git cat-file commit HEAD^{commit}  
  % git show --format=...
```

libgit.a

NOTE: This function isn't used in Git outside graph.c but it is used by CGit [...] to use HTML for colors.


```
die("foo is barred");
```



libgit2







libgit2

attr

[git_attr_add_macro](#), [git_attr_cache_flush](#), [git_attr_foreach](#), [git_attr_get](#), [git_attr_get_many](#)

blob

[git_blob_create_frombuffer](#), [git_blob_create_fromchunks](#), [git_blob_create_fromdisk](#),
[git_blob_create_fromworkdir](#), [git_blob_free](#), [git_blob_id](#), [git_blob_is_binary](#), [git_blob_lookup](#),
[git_blob_lookup_prefix](#), [git_blob_owner](#), [git_blob_rawcontent](#), [git_blob_rawsize](#)

branch

[git_branch_create](#), [git_branch_delete](#), [git_branch_foreach](#), [git_branch_is_head](#), [git_branch_lookup](#),
[git_branch_move](#), [git_branch_name](#), [git_branch_remote_name](#), [git_branch_set_upstream](#),
[git_branch_upstream](#), [git_branch_upstream_name](#)

checkout

[git_checkout_head](#), [git_checkout_index](#), [git_checkout_tree](#)

clone

[git_clone](#)

Func

[attr](#) (5

[blob](#) (

[branch](#)

[checko](#)

[clone](#)

[commi](#)

[config](#)

[cred](#) (

[diff](#) (2

[giterr](#)

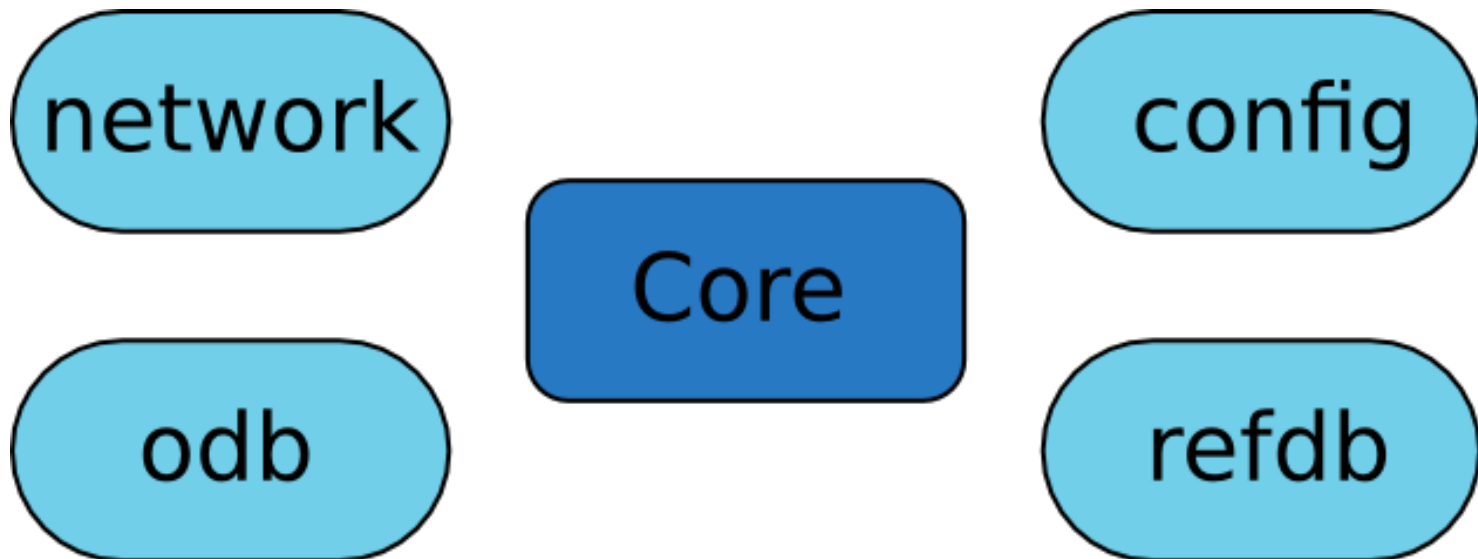
[graph](#)

[ignore](#)

versions

v0.13.0 v0.14.0 v0.15.0 v0.16.0 v0.17.0 v0.18.0 HEAD

Defined in: [blob.h](#)




```
git_repository_init(&repo,  
                  "projects/foo");  
  
git_repository_open(&repo,  
                  "projects/foo.git", 1);
```

```
git_reference_lookup(&ref,  
                    repo, "HEAD");
```

```
git_reference_lookup_resolved(&ref,  
                              repo, "HEAD");
```

```
git_branch_lookup(&ref, repo,  
    "master", GIT_BRANCH_LOCAL);  
  
git_branch_lookup(&ref, repo,  
    "origin/master",  
    GIT_BRANCH_REMOTE);
```

```
git_repository_config(&config, repo);  
  
    git_config_open_level(&cfg, config,  
        GIT_CONFIG_LEVEL_SYSTEM);
```

```
git_repository_config(&config, repo);  
  
git_config_open_level(&cfg, config,  
    GIT_CONFIG_LEVEL_SYSTEM);  
  
git_config_open_global(&cfg, config);
```

```
git_repository_odb(&odb, repo);  
  
    git_odb_add_backend(odb,  
                        my_fancy_odb, prio);
```

```
git_remote_set_transport(remote,  
                           my_cool_transport);
```

```
git_remote_set_callbacks(remote,  
                          &callbacks);
```

```
git_remote_stats(remote);
```

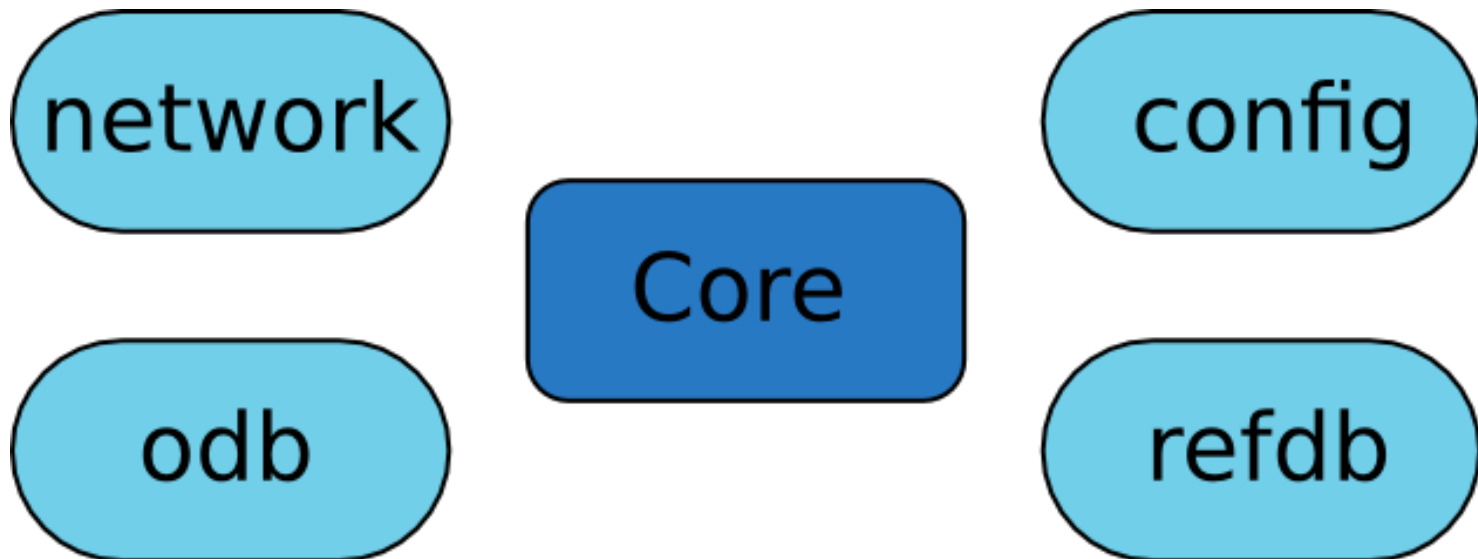


```
git_revparse_single(&obj, repo,  
                    "HEAD^2^{tree}");
```

```
git_revparse(&spec, repo,  
            "HEAD...HEAD^3");
```

```
git_status_file(&flags, repo,  
               "src/main.c");
```

```
git_status_foreach(repo, fun, NULL);
```



```
repo = Rugged::Repository.new(REPO_PATH)

get '*' do |path|
  commit = repo.lookup(repo.ref(REF_NAME).target)
  path.slice!(0)
  path = 'index.html' if path.empty?

  entry = commit.tree.path path
  blob = repo.lookup entry[:oid]
  content = blob.content
  halt 404, "404 Not Found" unless content

  content_type mime_type_for(path)
  content
end
```



GitHub



- Ruby - rugged
- Python - pygit2
- C# - LibGit2Sharp
- Objective-C - objective-git
- Go - git2go
- GLib - libgit2-glib

- Merge
- Event better API
- HFS+
- Sequencer
- Reflog backend

- http://en.wikipedia.org/wiki/File:Peter_Behrens_Porcelain_plate_line_ar_pattern_1901_BM.jpg
- http://upload.wikimedia.org/wikipedia/commons/0/07/ASU_Turbo-train.jpg

- * <https://libgit2.github.com/>
- * <https://github.com/libgit2/>
- * <https://github.com/libgit2/libgit2sharp/pull/429>
- * <https://github.com/fpco/gitlib/tree/master/gitlib-s3>



Thank you for your attention!

Questions?