

# Qordoba CLI v4

## Re-Building using Go programming language for mainly 5 reasons

Compiles to a single static binary

```
GOOS=windows go build -o cli.exe
GOOS=linux go build -o cli
GOARCH=armv7 GOOS=linux go build -o cli-rpi
```

Easy to create a REST client (Go includes a no-nonsense http client and has built-in support for working with xml, json and binary formats.)

Fast on every platform

Integrate with package managers

The standard Go library includes a flags package that can be used to parse flags or arguments in a couple of lines of code. (Go is unpretentious - you can have a single file that compiles to a tiny binary and is done with that.)

## Command Reference

### 1. Push files (Support file versions)

```
qor push
Push all local source files to the Qordoba project.

*files
(Optional). Lists the file paths to upload.
--update
(Optional). Updates an existing file.
--version
(Optional). Sets the version tag. Updates the file with version tag. Uploads a file with the
version tag
```

### 2. Push folders (Support file versions)

```
qor push
Push all local source files to the Qordoba project.

*files
(Optional). Lists the file paths to upload.
--update
(Optional). Updates an existing file.
--version
(Optional). Sets the version tag. Updates the file with version tag. Uploads a file with the
version tag
```

### 3. Pull file (Support file versions - support original source file)

```
Option
Description
-w --workflow
(Optional) Allow to download not completed translations. Will ask for feedback on workflowstep. By default - download only completed
translations.
-l --languages
(Optional) Option to work only on specific (comma-separated) languages
Examples:
```

```
./qor -l ru,en
./qor -lzh-cn
qor -len_US,ru,zh-cn
```

-f --force

(Optional) Force to update local files by remote translations. Do not ask approval.

--skip

Skip downloading if file exists. (default: False)

--replace

Replace existing file. (default: False)

--set-new

Ask to set new filename if file exists. (default: False)

--bulk

Speeds to the pull command:

Pulls translated files in bulk, including the source language. Will automatically create a directory 'bulk\_download' and store all translations files plus source files in it. No custom configuration of pull pattern possible

#### 4. Pull files (Support file versions - support original source file)

Option

Description

-w --workflow

(Optional) Allow to download not completed translations. Will ask for feedback on workflowstep. By default - download only completed translations.

-l --languages

(Optional) Option to work only on specific (comma-separated) languages

Examples:

```
./qor -l ru,en
./qor -lzh-cn
qor -len_US,ru,zh-cn
```

-f --force

(Optional) Force to update local files by remote translations. Do not ask approval.

--skip

Skip downloading if file exists. (default: False)

--replace

Replace existing file. (default: False)

--set-new

Ask to set new filename if file exists. (default: False)

--bulk

Speeds to the pull command:

Pulls translated files in bulk, including the source language. Will automatically create a directory 'bulk\_download' and store all translations files plus source files in it. No custom configuration of pull pattern possible

#### 5. Update file (Support file versions)

Option

Description

-w --workflow

(Optional) Allow to download not completed translations. Will ask for feedback on workflowstep. By default - download only completed translations.

-l --languages

(Optional) Option to work only on specific (comma-separated) languages

Examples:

```
./qor -l ru,en
./qor -lzh-cn
qor -len_US,ru,zh-cn
```

-f --force

(Optional) Force to update local files by remote translations. Do not ask approval.

--skip

Skip downloading if file exists. (default: False)

--replace

Replace existing file. (default: False)

--set-new

Ask to set new filename if file exists. (default: False)

--bulk

Speeds to the pull command:

Pulls translated files in bulk, including the source language. Will automatically create a directory 'bulk\_download' and store all translations

files plus source files in it. No custom configuration of pull pattern possible

## 6. Delete (Support file versions)

```
$ qor delete file_name.doc --version v1
```

## 7. Is

ID	NAME	#SEGMENTS	UPDATED_ON	STATUS
728068	runtime.yaml	6	2017-01-20 10:19:27	Enabled
725570	TestFileMH.json	26	2016-12-21 22:40:13	Enabled

## 8. Status per project or file (Support file versions)

```
$ qor status
```

#AUDIENSES	#WORDS	#SEGMENTS	EDITING	PROOFREADING	COMPLETED
ja-jp	94	32	0%	78.13%	21.88%
es-es	94	32	100%	0%	0%

```
$ qor status --json
```

```
{
  "command":
    Unknown macro: { "value"}
,
  "audienses_id": 1,
  "audienses_code": "A green door",
  "audienses_namer": "A green door",
  "audienses_code": "A green door",
  "audienses_code": "A green door",
  "audienses_code": "A green door",
  "audienses_name": 12.5
}
```

```
$ qor status file_name.docx
```

FILE NAME	#WORDS	#SEGMENTS	EDITING	PROOFREADING	COMPLETED
file_name.docx	94	32	0%	78.13%	21.88%

```
$ qor status file_name.docx --json
```

```
{
  "command":
    Unknown macro: { "value"}
,
  "audienses_id": 23,
  "audienses_code": "en-US",
  "filename": "file_name.docx",
  "file_id": 123442,
  "words": 94,
}
```

```
"segments": 32,  
  "status": "completed"  
}
```

9. Init

10. **Add key**

```
$ qor add-key file_name.doc --version v1 --key "/go_nav_menu" --value "text  
text text" --ref "Main nav text"
```

11. **Update value by key**

```
$ qor update-value file_name.doc --version v1 --key "/go_nav_menu" --value  
"text text text" --ref "Main nav text"
```

12. **Delete key**

```
$ qor delete-key file_name.doc --version v1 --key "/go_nav_menu"
```

13. **Create content**

```
$ qor new file_name  
> Text editore.....
```

14. **Pull value by key**

```
$ qor value-key file_name.doc --version v1 --key "/go_nav_menu"  
>  
+-----+-----+-----+-----+-----+-----+  
| FILE NAME | #VERSION| #KEY | #VALUE | #REF | #TIMESTAMP |  
+-----+-----+-----+-----+-----+-----+  
| file_name.docx | v1. | /go_nav_menu | text text text | Main nav text | 01/23/2019 @ 9:59pm (UTC)  
|  
+-----+-----+-----+-----+-----+-----+
```

15. **Score per file**

```
$ qor score file_name.doc --version v1
```

16. Local & file extensions code mapping support between source and target

```
qordoba:
access_token:
organization_id: 1234
workspaces_id: 12321
parallel: true
audiences_map: [ "123=EN_us", "12=EN_uK", "45=frFFrfs" ]

push:
sources:
- file: "./.lproj/Localizable.strings"

pull:
targets:
- file: "<audience_code>/<filename>.<extension>"

blacklist:
sources:
- file: "<audience_code>/<filename>.<extension>"
```

## 17. Results as JSON format

**Adding --JSON front of any command will give you the response as JSON**

### 18. Parallel push/pull support

Add `parallel: true` #default false

### 19. Blacklist option

```
blacklist:
sources: - file: "<audience_code>/<filename>.<extension>"
```

## 20. Help

**\$ qor -h**

**\$ qor -help**

## 21. Version

**\$ qor -v**

**\$ qor -version**

**>Qordoba Cli v4.0**