# buf Protobuf toolkit

## CONCEPTS

buf reads buf.yaml for modules, lint and breaking rules, and buf.gen.yaml for code generation. Inputs default to the current directory.

**proto files** buf.yaml module
↓ `buf lint / format`
**Checked source** style ok
↓ `buf breaking`
**Compatible** vs. last release
↓ `buf generate`
**Generated code** buf.gen.yaml

## SETUP AND CONFIG

**`buf --version`**
Print the installed version

**`buf config init`**
Create a buf.yaml here

**`buf <command> --help`**
Help for any command

## BUILD

**`buf build`**
Compile all protos in the module

**`buf build -o image.binpb`**
Write a binary image

## LINT

**`buf lint`**
Lint the current module

**`buf lint --path <file.proto>`**
Lint selected files only

**`// buf:lint:ignore <RULE_ID>`**
Silence one rule above a line

## FORMAT

**`buf format -w`**
Rewrite files in place

**`buf format -d`**
Show a diff instead

## FORMAT (cont.)

**`buf format -d --exit-code`**
Fail if anything is unformatted (CI)

## BREAKING CHANGES

**`buf breaking --against '.git#branch=main'`**
Compare with main in git

**`buf breaking --against <dir>`**
Compare with another directory

**`buf breaking --against image.binpb`**
Compare with a saved image

**Breaking categories**

| Category | Catches changes that break |
|---|---|
| `FILE` | Generated source per file (default) |
| `PACKAGE` | Generated source per package |
| `WIRE_JSON` | Wire and JSON encoding |
| `WIRE` | Wire encoding only |

## GENERATE

**`buf generate`**
Run plugins from buf.gen.yaml

**`buf generate --template <file.yaml>`**
Use another template file

**`buf generate --path <dir>`**
Generate for part of the module

## DEPENDENCIES AND MODULES

**`buf dep update`**
Resolve deps and write buf.lock

## CONVERT AND CALL

**`buf curl --list-services <url>`**
List services via reflection

**`buf curl -d '<json>' <url>/<pkg.Svc>/<Method>`**
Call an RPC with a JSON body

**`buf curl --schema <dir> <url>/<pkg.Svc>/<Method>`**
Use local protos, no reflection