

# Release workflow

## Pre release

- [Tests](#), check coverage
- [Mypy](#) check
- [Pylint](#) check
- [Translations](#)

```
pybabel extract -F openatlas/translations/babel.cfg -k lazy_gettext -o openatlas/translations/
messages.pot .
pybabel update -i openatlas/translations/messages.pot -d openatlas/translations
```

- Update version in config.py
- Update database version (if needed) in install and upgrade SQL
- Check update SQL
- Update model image in application, Sphinx and Redmine
- [Sphinx](#): update manual, version, check links with uncommenting print in `util/util.py manual()`

```
sphinx-build ./sphinx/source/ openatlas/static/manual
```

- Update changelog, release date
- Update upgrade and installation notes
- Prepare newsletter, news

## Release

- Git branches
  - merge **release\_candidate** to **main**
  - remove **release\_candidate**
  - merge **main** to **develop**
  - raise version in **develop**
- Push to GitHub
- Upgrade demo versions, if database upgrades:

```
pg_dump openatlas_demo > /var/lib/postgresql/openatlas_demo.sql
pg_dump openatlas_demo_dev > /var/lib/postgresql/openatlas_demo_dev.sql
```

- Test demo
- Post news
- Upgrade CRAWs, ACDH-CH and NHM instances
  - Download new database dump
  - Upgrade code and database
  - Send newsletter
  - Raise version in the online versions list

## Post release

- Update Redmine roadmap
- Add news at openatlas.eu
- Git tag
- Push main and tags to GitHub, GitLab and Bitbucket
- Update GitHub latest release
- Update stats in wiki