
bel Documentation

Release 0.12.17

William Hayes, David Chen

Jan 01, 2019

Contents:

1	Informational badges	3
1.1	Overview	3
1.2	Configuration	3
1.3	Installation	7
1.4	BEL CLI	8
1.5	Modules	8
1.6	Glossary	8
1.7	Contributing	8
1.8	Code of Conduct	9
1.9	DevOps	10
1.10	Authors	11
2	Related Documentation	13

Welcome to the *BEL Python Package* documentation.

1.1 Overview

Note: Namespace values (NSArg) need to be quoted if they contain whitespace, comma or ‘)’. This is due to how BEL is parsed. An NSArg (namespace:term, e.g. namespace argument of a BEL function) is parsed by looking for an ALLCAPS namespace prefix, colon and then a term name. The parsing continues for the term name until we find a space, comma or end parenthesis ‘)’. If the term contains any of those characters, it has to be quoted using double-quotes.

Note: Any character except an un-escaped double quote can be in the NSArg if it is quoted including spaces, commas and ‘)’.

1.2 Configuration

All of the BELbio tools share the same configuration files. There are separate sections for configuration that is specific to a particular application or tool (e.g. `bel_api`, `bel_resources` or `bel_python` package).

1.2.1 Location

Get first `belbio_conf.{yaml|yml}` and `belbio_secrets.{yaml|yml}` files in:

1. `<current dir>/belbio_{conf|secrets}.{yaml|yml}`
2. `ENV['BELBIO_CONF']/belbio_{conf|secrets}.{yaml|yml}`

3. ~/.belbio/{conflsecrets} (dotfiles in home directory)

Download example files from:

- BELBio Conf
- BELBio Secrets

1.2.2 Main Configuration File

```
# YAML Configuration file for BELbio tools

# Location of this file can be found in one of the following locations
# first found is the one that is used
# - ./belbio_conf.yml      (just a regular file in the local directory or
#                          any parent directory)
# - ~/.belbio_conf        (a dotfile in the home directory)

# bel package settings
bel:
  lang:
    default_bel_version: 2.0.0

    # Location of BEL Language Specification yaml files for different versions
    specifications: <dir_name>

    # Collect necessary files from belbio/bel_specifications repo if True
    specification_github_repo: True

    # Term Canonicalization and decanonicalization
    # Note that we don't decanonicalize CHEBI (this is not necessarily a symmetric_
    ↪process)
    canonical:
      HGNC: ["EG", "SP"]
      MGI: ["EG", "SP"]
      RGD: ["EG", "SP"]
      ZFIN: ["EG", "SP"]
      SP: ["EG"]
      ChEMBL: ["CHEBI"]

    decanonical:
      # EG will convert into the first valid namespace based on species
      EG: ['HGNC', 'MGI', 'RGD', 'ZFIN', 'SP']

  nanopub:
    # JSON Schema for BEL Nanopubs (in YAML format :)
    schema_uri: https://raw.githubusercontent.com/belbio/schemas/master/schemas/
    ↪nanopub_bel-1.0.0.yaml

    # Rules to apply for computing BEL Edges from BEL Nanopubs
    # pipeline_edge_rules: ["component_of", "degradation"]
    pipeline_edge_rules: [] # If empty, don't filter out any of the compute rules

# bel_api service settings
bel_api:
```

(continues on next page)

(continued from previous page)

```

# Should we require authentication in API?
authenticated: false

# Log configuration is in the conf-logging.yml file

# For accessing localhost from docker on a mac, use special dns name 'docker.for.
↪mac.localhost'
servers:
  server_type: DEV

  # Require JWT token for authentication using shared_secret from belbio_secrets.
↪conf?
  # HTTP Header: Authorization: JWT xxxxxx.yyyyyyy.zzzzzz
  authenticated: False

  # api_url: http://localhost:8181
  api_url: https://api.bel.bio/v1

  # elasticsearch: http://docker.for.mac.localhost:9200
  elasticsearch: http://localhost:9200

  # arangodb_host: docker.for.mac.localhost
  arangodb_host: localhost
  arangodb_protocol: http
  arangodb_port: 8529
  arangodb_username: ''
  # arangodb_password - comes from secrets file - will be merged in as config[
↪'secrets']['bel_api']['servers']['arangodb_password']

bel_resources:

  # restrict collecting to listed species, empty array means collect everything
  # species_list: ['TAX:9606', 'TAX:10090', 'TAX:10116']
  species_list: []

  # Will download new source datafiles if older than 7 days if we cannot determine the
  # source file modification date (some FTP sites or Web files make it hard to
↪figure
  # out when they were last modified).
  update_cycle_days: 7

  # Everything is relative to bel_resources root folder unless it starts with '/'
  file_locations:

    # Root filepath defaults to the repository root with the tools and data sub-
↪directories
    root: ''

    # Downloads - location for all source datafiles to be stored
    downloads: downloads
    # Data files - converted from source databases into standard formats
    data: data
    # Resources - static resources deployed with repository
    resources: resources
    # Files converted and ready to load into databases
    staging_files: staging

```

(continues on next page)

(continued from previous page)

```
# Term collection scripts
terms_data: "data/terms"

tools: tools

# Namespaces metadata definition file
namespaces_definition: resources/namespaces.yaml

# OpenBel related resource configuration - these will eventually get converted
# to native BELBio scripts as they don't provide for equivalents and other
# enhancements to the namespaces available in BELBio
openbel:
  # Used to select what to convert using
  # github:belbio/bel_resources/tools/terms/openbel_annotations.py
  openbel_annotation_sources:
    Anatomy: http://resources.openbel.org/belframework/20150611/annotation/anatomy.
↪belanno
    Cell: http://resources.openbel.org/belframework/20150611/annotation/cell.belanno
    CellLine: http://resources.openbel.org/belframework/20150611/annotation/cell-
↪line.belanno
    Eco: http://belief-demo.scai.fraunhofer.de/openbel/repository/annotation/
↪evidence.belanno

  # Used to select what to convert using
  # github:belbio/bel_resources/tools/terms/openbel_namespaces.py
  openbel_namespace_sources:
    AFFX: http://resources.openbel.org/belframework/20150611/namespace/affy-
↪probeset-ids.belns
    # LMSD: http://belief-demo.scai.fraunhofer.de/openbel/repository/namespaces/
↪LMSD.belns
    PMIBP: http://belief-demo.scai.fraunhofer.de/BeliefDashboard/dicten/namespaces/
↪pmibp.belns
    PMICHEM: http://belief-demo.scai.fraunhofer.de/BeliefDashboard/dicten/
↪namespaces/pmichem.belns
    PMICOMP: http://belief-demo.scai.fraunhofer.de/BeliefDashboard/dicten/
↪namespaces/pmicomcomp.belns
    PMIDIS: http://belief-demo.scai.fraunhofer.de/BeliefDashboard/dicten/namespaces/
↪pmidis.belns
    PMIPFAM: http://belief-demo.scai.fraunhofer.de/BeliefDashboard/dicten/
↪namespaces/pmipfam.belns
    SCHEM: http://resources.openbel.org/belframework/20150611/namespace/selventa-
↪legacy-chemicals.belns
    SCOMP: http://resources.openbel.org/belframework/20150611/namespace/selventa-
↪named-complexes.belns
    SDIS: http://resources.openbel.org/belframework/20150611/namespace/selventa-
↪legacy-diseases.belns
    SFAM: http://resources.openbel.org/belframework/20150611/namespace/selventa-
↪protein-families.belns
    ZFIN: http://belief-demo.scai.fraunhofer.de/openbel/repository/namespaces/zfin-
↪zebrafish-genes.belns

# Controls logging for all tools
logging:
  version: 1
  disable_existing_loggers: False
  formatters:
    simple:
```

(continues on next page)

(continued from previous page)

```

    format: '%(levelname)s::%(name)s::%(asctime)s   %(message)s'
handlers:
  console:
    class: logging.StreamHandler
    level: INFO
    formatter: simple
    stream: ext://sys.stdout
  # logfile:
  #   class: logging.FileHandler
  #   formatter: simple
  #   filename: '../api.log'
loggers:
  elasticsearch:
    level: WARNING
  urllib3:
    level: CRITICAL
root:
  level: INFO
  handlers: [console]
  # handlers: [console, logfile]

```

1.2.3 Secrets Configuration File

```

# belbio secrets - added to config['secrets']

# Location of this file can be found in one of the following locations
# first found is the one that is used
# - ./belbio_secrets.yml      (just a regular file in the local directory
#                             or any parent directory)
# - ~/.belbio_secrets        (a dotfile in the home directory)

bel_api:
  shared_secret: testme # shared secret to validate JWT token
  servers:
    arangodb_password: ''

```

1.3 Installation

1.3.1 Users

bel works on Python versions 3.6+. Is supported on OSX and Linux and works (but tested less frequently) on Windows.

Installing is simple.

```
pip install bel
```

1.3.2 Developers

If you are a developer, please see the *Contributing* first and then:

```
git clone <your forked version of bel>
```

The following commands sets up a virtual environment, installs *bel* as an editable package and then pip installs the requirements.txt packages.

```
make dev_install
```

To run the tests

```
make tests
```

1.4 BEL CLI

The BEL CLI commands are installed by `pip install bel`. These are nested commands for which you can review the help at each nested level by:

belc stands for BEL CLI (Command Line Interface) .. code-block:: bash

```
belc -help belc stmt -help belc stmt canonicalize -help
```

1.5 Modules

1.5.1 bel.lang.bel module

1.5.2 bel.lang.ast module

1.5.3 bel.lang.bel_specification module

1.5.4 bel.lang.bel_utils module

1.5.5 bel.lang.computed_edges module

1.5.6 bel.lang.semantics module

1.6 Glossary

This is maintained in the BEL.bio API documentation

[BELBio Glossary](#)

1.7 Contributing

Important: First off, thanks for taking the time to contribute!

The following is a set of guidelines for contributing to BEL.bio and its repositories, which are hosted in [BELbio](#) on GitHub. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request. If you have any questions, please let us know.

You will need to sign a Contributor's License Agreement (CLA) the first time you submit a pull request and anytime afterwards that the CLA agreement is updated (very rarely). This is signed by clicking on the request as part of the pull request and digitally signed with your Github ID automatically.

1.7.1 Code of Conduct

This project and everyone participating in it is governed by the *Code of Conduct Code of Conduct*. By participating, you are expected to uphold this code. Please report unacceptable behavior to one of the project maintainers, [William Hayes](#) or [Anselmo Di Fabio](#).

1.7.2 Code Contributions

This project uses the Git Forking workflow as discussed here <https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>

Tip: It may be helpful to review your plans with the community before starting work.

1. Please add an issue to the repository prior to working on a feature or bug.
2. Fork and clone the repository
3. Create a topic branch
4. After completing your changes, please ensure that the code style outlined in [Editorconfig](#) is followed.
5. Submit a pull request referencing the issue being resolved
 - Other examples of Git Forking workflow <http://www.asmeurer.com/git-workflow/> or <http://blog.scottlowe.org/2015/01/27/using-fork-branch-git-workflow/>

1.8 Code of Conduct

Contributor Covenant Code of Conduct

1.8.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

1.8.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community

- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

1.8.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

1.8.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

1.8.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [William Hayes](#) or [Anselmo Di Fabio](#). All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

1.8.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](http://contributor-covenant.org/version/1/4), version 1.4, available at <http://contributor-covenant.org/version/1/4>

1.9 DevOps

All administrative tasks are managed by running make tasks using the top-level Makefile in the project folder.

1.9.1 Builds and Testing

We use TravisCI for Open Source to run builds and tests.

1.9.2 Documentation

Documentation is based on Sphinx and created and hosted at [ReadTheDocs](#) using the *belbio* user id.

1.9.3 Dependabot

We use <https://app.dependabot.com/accounts/belbio/repos> to keep the python module requirements up to date. It uses the *belbio* user id.

1.9.4 Code Quality

We are using [Code Climate](#) for code quality assessments.

We are using [CodeCov](#) for code test coverage assessments.

1.9.5 Contributor Licensing Agreements

All pull requests require signing the [CLA Assistant](<https://cla-assistant.io/>) Contributor's License Agreement.

1.10 Authors

The *bel* python package is written and maintained by the following:



William Hayes, @wshayes



David Chen, @davidchen

CHAPTER 2

Related Documentation

- [BEL API documentation](#) - BEL.bio REST API documentation
- [BEL Resource Tools documentation](#) - BEL resource generation (namespaces, orthologies, etc)