

api-to-roefact by RENware Software Systems

- [CHANGELOG 0.4.1.dev0](#)
 - [0.4.1.dev0 xl2roefact include a data directory in package for various data files "built-in" package \(240318 h09:00\)](#)

CHANGELOG 0.4.1.dev0

0.4.1.dev0 xl2roefact include a data directory in package for various data files "built-in" package (240318 h09:00)

- 240318piu-yaml4sys-all-vers update mkdocs.yml use INHERIT: ./versions.yml option to inherit versions exactly to have extra section like actual one but in this external file.
- 240318piu-app-glob-cfg update xl2roefact README, app configuration section ref how make a "global config file" different than default app configuration
- 240317piu01 update downloads.md:
 - ✓ 1. include a template for app_settings.yml => Created dedicated section which can be referred as `.../downloads.md#sablon-fisier-configurare-a-aplicatiei-xl2roefact`
 - ✓ 1.a fixed 0.4.0.dev2 MSI & EXE deliverables file names (missing path number)
 - ✓ 2. refer it in xl2roefact README
 - ✓ 3. include 0.4.1.dev download;
 - WHEEL & DIST references
 - standalone EXE reference
 - MSI reference
- 2403piu-config-code update config_settings.py module to upload data from `.../data/app_settings.yml` file. Specs:
 - ✓ 1. update app version to 0.4.1.dev0 . Make a test what `normalized_version()` returns (run `xl2roefact --version`)
 - ✓ 2. update key `README_rules` from `app_settings.yml` , set it to point to markdown file containing rules list (`xl2roefact/xl2roefact/data/README_app_config_rules.md`)
 - INFO-NOTE: after YAML import data will be dict with all actual code variables as keywords
 - ✓ 3. gross-raw code in `config.settings.py` for both below NOTES
 - INFO-NOTE: order to search and load for `app_config.yml`:
 - (1) crt directory (with `cwd`) with `Path(Path.cwd(), "data/app_settings.yml")`

- (2) package directory and file with `Path(os.path.dirname(__file__), "data/app_settings.yml")`
- (3) settings from `config_settings.py`
- INFO-NOTE: methods of updates variables:
 - (1) using `locals().update(YAML_dict)`
 - (2) using `exec(YAML_dict["key"])` by looping YAML resulted dictionary
- regardless of method check the propagation running `x12roefact settings` which is a demo of using values external to `config_settings` module
- ✓ 4. apply reading order
- ✓ 5. clean code & test YAML settings files from `x12roefact/`
- ✓ 6. if read YAML was got some values then set as `local()` variables (not dictionary)
- ✓ 7. updated `x12roefact/setup.py` to include `data/app_settings.yml` file
- ✓ build all: upd version (2 version files + `mkdocs.yml`) & `pdm run build_all`
- 2403piu-app-data-dir actions:
 - ✓ 1.a build directory with a TOML file for setting parameters (used by `config_settings` module)
 - ✓ 2 update `pyproject.toml` to include in package non python data files from `x12roefact/data/` directory
 - ✓ 2.a test `pdm` building wheel ref brute errors = **PASS** =: package created ok and contains `data/*` with exact files that exists in this directory at package development phase
 - ✓ 3. add in `../x12roefact/data/` file `owner_data.json` with owner data to be used as supplier info for future option `--load-from-owner-file`
 - ✓ 4. fixed bug `x12roefact` CLI app ref command `about` printing `__version__` & `__doc__` addressing
 - ✓ 5. updated `../data/app_settings.yml` with actual existing config data. Not usable as is, need refining and clarify how to indicate data types to app users (actually indicated as Python type hints)

Last update: April 19, 2024