xl2roefact component

This component is presented in document `.../doc_src/...110-SRE-api_to_roefact_requirements.md``

Detailed technical documentation

Component detailed specifications can be found in 810.05a-xl2roefact_DLD_specs document

Configuration & settings

Configuration constants and variables are placed in file **`config_settings.py`**. These are in Python form presented using constants PEP recommendations (all upper case) and accompanied by some help lines to understand and maintain them.

Working directories

- **`invoice_files/`** normal directory for Excel files to be processed (here will be searched Excel files if not foud in current directory)
- ___ `build/<source-file-name>/` directory which will contain intermediary files usable when rebuilding CLI application
- ***`dist/<source-file-name>.exe`*** executable generated file (format Windows x86)
- **`test_data_and_specs_originals/`** contains test invoices: from client, a RENware one, a 3rd party one:
  - **`specs/`** contains specification documents + `_my_notes.md` with notes & comments made in analysis phase
  - **`fact_*/`** original test invoices: from Kraftanlagen, from RENware, from 3rd party

Creating and deploying component

Building Windows executable

- Change to `base-proc/` directory
- Activate environment: `.\.venv\Scripts\activate`
- build packages with PDM:
  - **`pdm build_wheel`** build Python package,
  - **`pdm build_msi`** build MSI package,
  - **`pdm build_all`** build all packages
- build technical documentation
  - **`pdm build_doc`** build technical documentation. *NOTE-ATTN* name of markdown generated file

NOTE building packages options

In some cases may be useful to run basic-raw commands - here these are: * build MSI package: run **`python setup.py bdist_msi`** which will build: * **`EXE`** (`build/bdist.win-amd64/`) and * **`MSI`** Windows install package (`dist/xl2roefact-0.1-win64.msi`) * as result will be created following directories with the `Windows` files for deployment:

```
build/
    bdist.win-amd64/ - containing the msi type installer
    exe.win-amd64-3.10/ - containing the executable file(s)
dist/
    xl2roefact-0.1-win64.msi - is the Windows installer
```

* build PyPy package if needed: **`pdm build --no-clean`**

NOTE ref deployment to end users

- is recommended to assembly an archive file to be deployed as release package by `git` (anyway is recommended as executable and msi files could not be easily deployed or sent by e-mail)
- MSI package is deployable itself (as-is)

System modules

`xl2roefact` basic "active" modules (not package definition) are:

- `rdinv` read an Excel file and extract invoice data to a JSON file format
- `wrxml` write, convert the JSON invoice file to a XML file format, respecting schemes required by *RO EFact* standard

- `chkxml` check generated XML file
- `ldxml` load an invoice (ie, its XML associated file) to *ANAF SPV system*
- `chkisld` check if an invoice is already loaded in *ANAF SPV system*
- `config_settings` define system settings & parameters mainly used in invoice info / data detection and extract from invoice Excel format file
- `app_cli` contains the code for `xl2roefact` application command line (CLI) format

Below are presented the ***skeleton logic*** of those modules where is relevant, meaning is not enough obvious from code or code complexity exceed usual limits (*for example nore than 100 lines of code per function*). For technical details and specification regarding modules see 810.05a-xl2roefact_DLD_specs.md file

rdinv module logic

Main function of `rdinv` module is `rdinv(...)` which has the following logic sections which are in ***strict sequence in presented order***:

- *search of* `invoice_items_area` *subtable*. This area is expected to contain invoice lines and is "processed" first because it is more structured and easier to identify; after its identification the header area is considered upper of it and footer area below it
- *solve* `invoice_items_area` *in 2 step....* In this step the code-data-variables of items area will be initialized in order to hold information that will be found
- *localize and mark areas for...* section that follows natural the previous one by initializing code-data-variables forcheader and footer areas to hold their corresponding information
- *solve* `invoice_header_area` detailed initialize of header area code-data-variables
- `ReNaSt` *-RegNameStrategy* section that identify and extract the legal registered name of invoice customer
- ...wip...
- *section to (Excel data)--->(JSON) format preparation and finishing* section which prepare Excel original data found to be be saved as JSON as a more "electronic interchangeable" structure

...#TODO for other modules if needed...

Last update: January 16, 2024