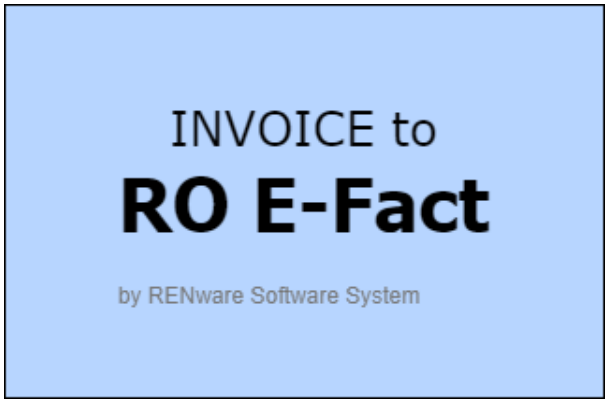
Table of Contents

II.III aplicatie desktop

II.IV aplicatie mobile

I. INVOICEtoROefact



RENware Software Systems

1 INVOICEtoROefact

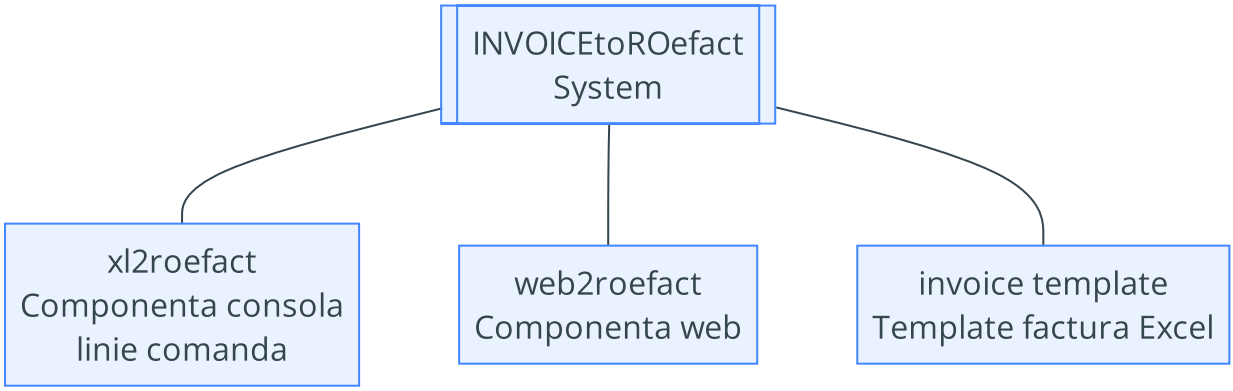
Facturi emise in Excel, problema rezolvata cu **INVOICEtoROefact** !

Emiteti si folositi in continuare facturi in Excel. Acest sistem va produce fisierul XML de care aveti nevoie pentru sistemul ANAF E-Factura (https://www.anaf.ro/anaf/internet/ANAF/despre_anaf/strategii_anaf/proiecte_digitalizare/e.factura) .

Si mai mult, daca aveti nevoie, puteti Integra aceste facturi si alte sisteme externe prin metode moderne, uzuale, curente, incetatenite de ani de zile si binecunoscute de specialistii din IT.

1.1 Componentele si facilitatile sistemului

Sistemul **INVOICEtoROefact** ofera urmatoarele componente:



- **xl2roefact** version 0.9.dev1 - procesarea facturilor din interfata linie de comanda (aplicatie tip consola)
  - **xl2roefact consola** aplicatie linie comanda CLI
  - **xl2roefact developer** biblioteca sursa python dezvoltari proprii

  - **xl2roefact desktop** ...in curind ... aplicatie desktop GUI
  - **xl2roefact mobile** ...in curind... aplicatie pentru mobile Android si iOS
- **web2roefact** version n/a - procesarea facturilor din interfara web
- **invoice template** version 0.1.20 - model / sablon factura in Excel

Descrierea facilitatilor acestor componente poate fi accessata aici.

1.2 Date identificare

- p/n: `0000-0095`
- code-name: `api_to_roefact`
- commercial name: **INVOICEtoROefact**
- site web (https://invoicetoroefact.renware.eu/)
- Git Hub repository (https://github.com/petre-renware/api_to_roefact)
- Git Hub releases (https://github.com/petre-renware/api_to_roefact/releases)
- copyright: RENware Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)
- general system license

II. xl2roefact

II.I aplicatie consola

RENware Software Systems

2 xl2roefact

Legaturi externe utile:

- **Web Site** (https://invoicetoroefact.renware.eu/) . (*Pentru acces corect la toate referintele din acest document vizitati site-ul dedicat acestui sistem.*)
- Pachet *PyPi* (https://pypi.org/project/xl2roefact/)
- Surse *GitHub* (https://github.com/petre-renware/api_to_roefact/)
- Referinta *dezvoltare software - biblioteca Python* (https://invoicetoroefact.renware.eu/xl2roefact/doc/README_xl2roefact_library.html)
- Referinta *API* (https://invoicetoroefact.renware.eu/xl2roefact/doc/wrapper_810.05a-xl2roefact_DLD_specs.html)

2.1 Facilitati

Aceasta componenta este "totul despre crearea de facturi electronice" din formatul Excel office (xlsx). Aplicatia poate genera factura in format JSON, XML, PDF si o poate incarca in sistemul *RO E-Fact*[1].

Aceasta componenta ofera urmatoarele facilitati (acestea fiind obiectivele fundamentale ale componentei):

- **transformarea facturilor din Excel in formatul** `XML` cerut de catre sistemul ANAF RO E-Fact pentru incarcare
- **incarcarea acestora** in sistemul ANAF RO E-Fact[1]
- **transformarea facturilor din Excel intr-un format** `JSON` intermediar, independent de platforma si care permite integrarea acestora cu alte sisteme (standard *REST*)
- **generarea facturii in format PDF** pentru transmiterea acesteia catre client, semnarea electronica, tiparirea si arhivarea acesteia in format fizic (in general manipularea facturii in format *"human readable"*)

Componenta ofera doua instrumente pentru realizarea si indeplinirea acestor obiective:

- `xl2roefact` o **aplicatie de tip linie de comanda** (disponibila pentru sistemele de operare Windows, Linux si MacOS)

- `xl2roefact PyPi` **blioteca standard Python** utilizabila pentru dezvoltari proprii in scopul extinderii altor sisteme existente (*custom development*)

2.2 Instalarea aplicatiei xl2roefact

Instalarea aplicatiei *xl2roefact* este disponibila in urmatoarele variante:

- pentru **Windows**:
  - `MSI` pachet instalare pentru *Windows*
  - `EXE` executabil *Windows in format "portabil" (un singur fisier)*
- pentru **Linux**:
  - *...in curind...* `DEB` pachet instalare pentru *Linux Debian*
  - *...in curind...* `RPM` pachet instalare pentru *Linux*
  - *...in curind...* `APPIMG` executabil *Linux in format "portabil" (un singur fisier)*
- pentru **Mac OS X**
  - *...in curind...* `DMG` pachet instalare pentru *MacOS* <!--
- ca **script Python** indiferent de sistemul de operare;
  - **Pachet Python (https://pypi.org/project/xl2roefact/)** biblioteca / libraria completa pe PyPi (inclusiv sursele)

> *Pentru acces la pachetele de instalare vezi sectiunea de descarcare resurse.*

Note: * *utilizarea ca script Python necesita existenta ca mediul* Python3 min 3.10 *sa fie instalat local* * numele pachetelor includ versiunea de aplicatie utilizata si sistemul de operare pentru care sunt disponibile * pentru echivalent utilizare *portabila pentru Linux* se poate instala biblioteca Python dupa care devine utilizabil scriptul Python "ca orice alta comanda Linux"

2.3 Configurarea aplicatiei xl2roefact

Parametrii de configurare aplicatiei se gasesc in fisierul `config_settings.py`. Acestia sunt sub elaborati in limbaj Python prin utilizarea conventiilor de constante conform recomandarilor PEP (numele capitatlizat) si sunt acompaniti de linii de explicatii privind aplicabilitatea lor.

Configurare aplicatiei se poate face interactiv si din aplicatie. Pentru a obtine help referitor la detaliile comenzi se va folosi

```
xl2roefact settings --help
```

Configurarile existente si regulile recomandate in configurarea aplicatiei se afiseaza folosind comanda:

```
xl2roefact settings --rules
```

2.3.1 Configurarea din fisier extern

Configurarea aplicatuiei se poate face si prin intermediul unui fisier extern numit "*sablon de configurare*" (*en: configuration template*). Sablonul permite configurarea aplicatiei prin modificarea fragmentelor de text care trebuiesc cautate in fisierul Excel pentru identificarea diverselor informatii aferente facturii.

Sablonul este in format YAML (https://yaml.org/) iar informatiile ce trebuiesc descrise sunt explicate individual in comentarii insotitoare. De asemenea este util a fi citite si recomandarile date in pagina de descriere a aplicatiei.

Pentru a beneficia de cobfigurarile facute de dumneavoastra trebuie sa creati un fisier `app_settings.yml` in directorul curent din care lansati aplicatia, fisier ce contine noile configurari dorite. **Numele fisierelui este obligatoriu a fi respectat.**

> **Fisiere de configurare multiple**
>
> De retinut ca acest fisier este considerat (daca exista) cel din directorul curent de unde lansati aplicatia. Deci daca v-ati creat mai multe directoare de lucru (de exemplu pentru clienti diferiti) puteti crea fisiere de configurare specifice, cite unul in fiecare director.

Fisier de configurare global

In conditiile folosirii kitului MSI pentru o instalare locala a aplicatiei (cu utilizari multiple si repetate) si in situatia in care se doreste schimbarea configurarii implicite a aplicatiei se vor urma acesti pasi:

- in directorul de instalare a aplicatiei se va crea daca nu exita directorul `data/`
- in acest director se va crea un fisier `app_settings.yml` cu configurarea globala dorita

Aceasta configurare inlocuieste configurarea implicita si se va aplica global in utilizarea aplicatiei. In continuare configurarile existente in directorul curent *suprascriu configurarea globala* (se aplica cu precedenta).

> Aici puteti gasiti pentru descarcare un model de sablon de configuare.

2.4 Utilizare nomenclator de furnizori

Aplicatia *xl2roefact* permite utilizarea datelor pentru furnizori din fisiere externe (in locul informatiilor din fisierele Excel) lucru ce poate fi folositor in urmatoarele situatii:

- cind utilizatorul aplicatiei o face in scopuri personale si multe facturi emise il au *pe el ca furnizor*. Aceast lucru permite ca informatia din Excel referitoare la furnizor sa fie sumara sau sa lipseasca, factura finala format `PDF` fiind generata cu aplicatia

- cind utilizatorul aplicatiei o foloseste pentru a emite facturi pentru alte firme si astfel este mai comod sa foloseasca fisiere cu datele acestor firme decit sa introduca informatia in fiecare factura
- cind se doreste ca datele furnizorului sa fie preluate dintr-un sistem extern ce le poate exporta ca si fisisre

2.4.1 Reguli generale de utilizare

Aceasta sectiune descrie regulile generale ce trebuiesc avute in vedere pentru o completa si corecta utilizare a facilitatii "Nomenclator furnizori":

- Nomenclatorul de furnizori se va completa intr-unul sau mai multe fisere de date (de tip text, vezi mai jos formatul exact).
- Un fisier acomodeaza un singur furnizor. Pentru mai multi furnizori se vor folosi fisiere diferite.
- Numele fisierului (fara extensie) trebuie sa coincida cu o cheie alternativa a furnizorului respectiv. Prin cheie alternativa se intelege acea cheie care este unica si poate asigura regasirea furnizorului prin folosirea ei. Ca si exemple din practicile curente ar fi cimpul numit uzual `code` sau `code_name` existent in mai toate sistemele de business. Acesta are avantajul unicitatii si a unei reprezentari "umane" (*en: human readable*). Desigur un cimp de tip cheie primara / ID este ideal dar de obicei acesta este tehnic iar valoarea sa nu ofera prea multe indicatii.
- Formatul fisierului este YAML (https://yaml.org/) standard, fara folosirea de modele de date complexe, aatfel incit o eventuala conversie `JSON` <--> `YAML` sa poata fi realizata manual in ambele sensuri si fara necessitatea unor cunostinte avansate ci la nivel de redefinire a numelor cheilor.

2.4.2 Locatia nomenclatorului

Fisierele cu datele furnizorilor pot sta in urmatoarele locatii:

- *directorul curent* este locatia cu prioritatea maxima si in caz de "duplicate" ale unui fisier, cel de aici va fi luat in considerare
- *directorul* `data/` din locatia unde este instalata aplicatia

Recomandari si practici uzuale:

- In situatiile in care sistemul este instalat pe un computer ce se foloseste frecvent cu aplicatia *xl2roefact* si exista un set de furnizori frecvent folositi se recomanda folosirea directorului `data/` pentru stocarea fisierelor nomenclator astfel incit sa poata fi refolosite usor.
- In situatia folosirii a "multe" fisiere date furnizori se recomanda crearea unui director dedicat in locatia utilizata (vezi mai sus) si acesta sa fie referit in numle fisierului.

2.4.3 Utilizarea nomenclatorului

Pentru a folosi cu aplicatia un fisier tip nomenclator furnizor se va utiliza optiunea:

```
xl2roefact xl2json -o fisier_furnizor
```

unde `fisier_furnizor` este numele fisierului ce contine datele unui furnizor. Locatia acestui fisier este relativa la locatia considerata pentru folosire

2.4.4 Sablon pentru nomenclator de furnizori

Sablonul este proiectat pentru utilizarea in facturile emise si contine numai informatiile necesare in acest scop.
Astfel cimpurile existente trebuiesc pastrate, adica nu vor fi sterse.

Vor fi *respectate si completate* corespunzator cimpurile specificate ca *obligatoriii (en: mandatory* in comentariile aferente fiecarui cimp.
Pentru acele cimpuri pentru care informatia este necunoscuta sau considerata irelevanta se va completa cu `null`.

Se vor putea adauga orice alte cimpuri suplimentare cu conditia sa fie respectat formatul fisierului (`YAML`). acestea nu vor fi folosite de catre aplicatie, ci pur si simplu ignorate.

> Aici puteti gasiti un model de sablon de configuare.

2.5 Comenzile aplicatiei

Interfata aplicatie este realizata utilizind conventiile si practicile uzuale pentru aplicatii tip linie de comanda consola. Pentru informatii privind comenzile se poate folosi optiunea de **help**, dispobilia atit la nivelul general:

```
xl2roefact --help
```

cit si la nivel detaliat pentru fiecare comanda

```
xl2roefact [COMMAND] --help
```

Lista comenzilor:

- **about** - Afiseaza informatii despre aceatsa aplicatie (copyright, scop, etc)
- **settings** _ Afiseaza parametrii de configurare a aplicatiei. Vezi sectiunea de configurare a aplicatiei
- **xl2json** - Transforma fisierul (fisierele) Excel in forma JSON pentru utilizare ulterioara ca forma de date standardizat pentru schimbul de informatii cu alte sisteme electronice

Comenzile detaliate:

2.5.1 xl2roefact

Application global information (command agnostic).

Usage:

```
xl2roefact [OPTIONS] COMMAND [ARGS]...
```

Options:

```
  -V, --version         show application version
  --install-completion  Install completion for the
                        current shell.
  --show-completion     Show completion for the
                        current shell, to copy it
                        or customize the
                        installation.
  --help                Show this message and
```

2.5.1.1 about

Provide a short application description.

Usage:

```
xl2roefact about [OPTIONS]
```

2.5.1.2 settings

Display application configuration parameters and settings that are subject to be changed by user.

Args:

```
`rules`: show recommended rules to follow when change application configurable settings
(available in both RO & EN languages). Defaults to `False`.
```

Usage:

```
xl2roefact settings [OPTIONS]
```

Options:

```
  -r, --rules  show settings recommended update
               rules
```

2.5.1.3 xl2json

Extract data from an Excel file (save data to JSON format file with the same name as original file but `.json` extension).

Args:

```
`invoice_type_code`: invoice type (for exaple regular invoice or storno) as this info is not
usually subject of Excel file. Default to `380` (regular / usual invoice)
`file_name`: files to process (wildcards allowed).
`files_directory`: directory to be used to look for Excel files. Defaults to `invoice_files/`.
NOTE: if default directory does not exists will consider current directory instead
`owner_datafile`: File to read invoice supplier (owner) data instead Excel.
`verbose`: show detailed processing messages". Defaults to `False`.
```

Usage:

```
xl2roefact xl2json [OPTIONS] [FILE_NAME]
```

Options:

```
  --invoice-type [380]            [default: 380]
  [FILE_NAME]                     files to process
                                  (wildcards
                                  allowed)
                                  [default:
                                  *.xlsx]
  -d, --files-directory DIRECTORY
                                  directory to be
                                  used to look for
                                  Excel files (if
                                  default
                                  directory does
                                  not exists will
                                  consider current
                                  directory
                                  instead).
                                  [default:
                                  invoice_files/]
  -o, --owner-datafile FILE       File to read
                                  invoice supplier
                                  (owner) data
                                  instead Excel.
  -v, --verbose                   show detailed
                                  processing
                                  messages
  --help                          Show this
                                  message and
```

2.6 Practici si regului referitoare la continutul facturilor din Excel

Acest capitol se refera la modul in care este "tratat" continutul fisierului Excel cu factura, mai exact la modalitatea in care informatia facturii este cautata, identificata si gasita in scopul de a fi salvata in oricare din formatele de "factura electronica / E-Fact".

Utilizarea sablonului de factura Excel ce este livrat impreuna cu aplicatia **ESTE O VARIANTA DE LUCRU RECOMANDATA**, dar nu obligatorie. Chiar si in cazul utilizarii acestuia, prin modificarea "structurii" acestuia, informatia poate ajunge *nerecognoscibila / neidentificabila* total sau partial daca nu sunt urmate regulile expuse.

> In general trebuie facuta diferenta intre datele facturii si modul in care aceasta va fi tiparita (va aparea la tiparire / previzualizare).
>
> Mai exact **continutul informational** al facturii nu trebuie nici confundat si nici mixat cu **formatul de afisare al acesteia** (layout). Pentru acesta din urma se recomanda a fi folosite cu precadere *regulile de formatare* din Excel si nu cele stocare a datelor. Un exemplu este un numar zecimal oarecare unde:
>
> - una este valoarea introdusa intr-o celula (de ex cu 3 zecimale) si
> - alta este valoarea afisata (cu 2 zecimale) - aceasta din urma trebuie obtinuta prin formatarea celulei respective de a afisa 2 zecimale prin rotunjire insa valoarea efectiva trebuie sa fie cea originala cu 3 zecimale, lucru (diferenta) care se poate vedea la editarea continutului celulei.

2.6.1 Reguli recomamdate in configurarea aplicatiei pe specificul Excel al facturilor dumneavoastra

Reguli recomandate pentru adaptarea aplicatiei la modelul dvs de factura in Excel:

- fiecare parametru are un hep scurt (liniile ce incep cu caracterul `#`) - citi-l inainte de a modifica uun parametru
- nu schimbati numele parametrilor asa cum este el specificat inainte de semnul egal (`=`)
- listele sunt incluse intre paranteze drepte (`[...]`) si elementele lor sunt separate prin caracterul virgula (`,`)
- sirurile de caractere sunt incluse intre ghilimele (caracterul `"`)
- daca doriti stergerea unei listei (de ex daca nu doriti nici o optiune pentru acea lista) doar lasati acel parametru cu valoarea `[]` - nu stergeti in nici un caz acel parametru
- nu adaugati parametrii suplimentari (altii decit cei specificati aici), acestia nu vor fi utilizati fara a modifica aplicatia (de asemenea riscati sa induceti erori in cod)
- pentru datele calendaristice in celulul Excel a se utiliza formatul standard de data (`date`) si modificati formatul de afisare in formatul dorit pe factura tiparibila

(en-us) Recommended rules to configure the application to your Excel invoice model:

- each parameter has a short help (lines starting with `#` character) - read it before changing that parameter
- do not change parametrs name as specified before equal (`=`) sign
- lists are enclosed in sqaured brackets (`[...]`) and items are separated by comma character (`,`)
- strings are enclosed in `"` characters
- if you want to clear a list (for example you do not wants any options inside) just let it as `<PARAMETR NAME> = []` - do not drop that parameter
- do not add supplementary parameters, they will not be used without software changes (also risk to induce potential errors)

- for calendaristic dates Excel cells use `date` format and change it as display option to show wanted format

2.7 Tutorial utilizare aplicatie

2.7.1 Organizarea informatiei

Aplicatia *xl2roefact* "promoveaza" structurarea informatiei procesate astfel incit sa fie evitata situatia *"de aglomerare" a directorului curent cu fisiere* ce trebuiesc identificate si izolate in situatia in care se fac *procesari in masa* (pe mai multe fisiere / facturi sursa).

Astfel, aplicatia se asteapa ca fisierele Excel sursa (*adica facturile de procesat*) sa fie copiate in directorul **`invoice_files/`** de unde vor fi citite si tot aici vor fi create fisierele rezultate (JSON, XML, etc). Acest director este relativ la directorul curent de unde este lansata aplicatia si considerat *"implicit"* cu acest nume dar daca se doreste un alt director acest lucru poate fi facut folosind parametrul `--files-directory` (sau prescurtat `-d`) la lansarea aplicatiei astfel:

```
xl2roefact -d "calea si numele directorului dorit"
```

> **Nota**
>
> Ghilimelele sunt necesare numai daca numele si calea (`path`) contin caracterul spatiu.

Exemple:

- pentru stabilirea directorului curent ca sursa pentru fisierele factura Excel:

  ```
  xl2roefact -d ./
  ```

- procesarea tuturor facturilor facturilor din luna *iunie*, copiate intr-un director dedicat sub directorul curent:

  ```
  xl2roefact -d ./facturi_iunie/
  ```

2.7.2 Exemplu de procesare a unei facturi

- se creaza directorul recomandat pentru stocarea facturilor in Excel:

  ```
  md invoice_files
  ```

- se copiaza factura `factura_A.xlsx` in acest director apoi se revine in directorul anterior daca acesta a fost schimbat pentru efectuarea copierii

- se lanseaza aplicatia:

```
xl2roefact xl2json
```

In urma acestor operatii, in directorul `invoice_files` vor rezulta:

```
 invoice_files/
├──  factura_A.xlsx  # fisierul Excel original
└──  factura_A.json  # fisierul JSON rezultat in urma procesarii
```

- `factura_A.xlsx` ca fiind fisierul Excel original cu factura
- `factura_A.json` acesta fiind fisierul format JSON rezultat in urma procesarii si ce poate fi folosit pentru interschimbarea electronica a informatiei intre sisteme

2.8 Aspecte tehnice referitoare la formatul fisierului JSON aferent facturii

Acest fisier este cel generat de catre aplicatie in urma executiei acesteia cu comanda `xl2json`. Formatul JSON are urmatoarra structura de baza:

```
{
    "Invoice": {...},
    "meta_info": {...},
    "excel_original_data": {...}
}
```

Cheile de la primul nivel contin:

- **`Invoice`** - datele efective ale facturii
- **`meta_info`**
  - informatii referitoare la procesarea facturii si mapa de conversie a cheii `Invoice` din formatul `JSON` in formatul `XML` cerut de sistemul *RO E-Fact*
  - harta de ajutor in conversia formatului JSON in formatul XML acceptat de sistemul RO E-Fact (cheie `meta_info.map_JSONkeys_XMLtags`) si definititiile XML aferente (cheie `meta_info.invoice_XML_schemes`)
  - alte informatii despre fisierul Excel prelucrat (numele, worksheet cu factura, data si ora procesarii, CRC pentru verificare, etc)
- **`excel_original_data`** - informatiile originale din fisierul Excel, asa cum au fost ele identificate si gasite precum si locatia (adresele celulelor). Aceste informatii sunt utile in cazul in care exista neclaritati in urma procesuluicde conversie pentru "a intelege" de unde si cum arata informatiile originale din fisierul Excel

Detalii suplimentare despre formatul JSON se gasesc in documentaţia *Referinta dezvoltare software*.

2.9 Referinta dezvoltare software

Documentatia **"Referinta dezvoltare software"** ofera detail necesare pentru utilizarea bibliotecii sursa in dezvoltari specifice, extindere si integrare cu alte sisteme.

2.10 Date identificare

- part number (p/n): `0000-0095-xl2roefact`
- producator si copyright: RENWare Software Systems (referinte detalii tehnice: Petre Iordanescu, *petre.iordanescu@gmail.com*)

2.11 License

2.12 Note

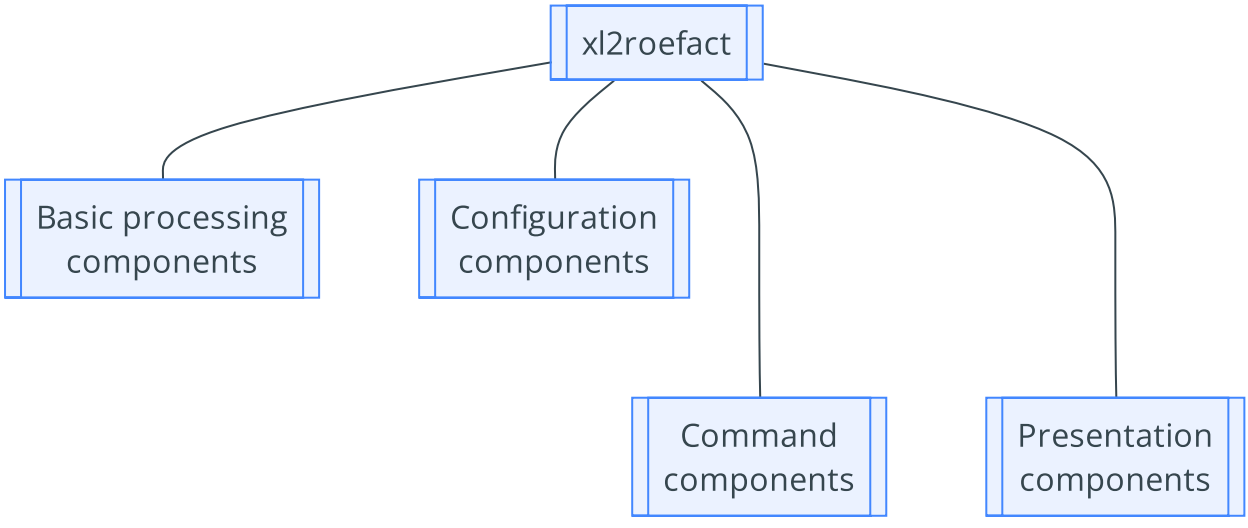
1. Toate interactiunile cu sistemul *ANAF RO E-Fact* necesita o *conexiune la internet* si un set de *credentiale ANAF RO E-Fact ale companiei* pentru care se incarca factura. In lipsa acestora, fisierul `XML` generat de aplicatie poate fi incarcat ulterior (de ex de catre departmentul contabilitate) ↩↩

II.II componenta developer

3 Software development reference using xl2roefact python library

version 0.9.dev1

3.1 Library components



3.1.1 Basic processing components

These components assure the basic elementary-raw processing of information. Their interface is pure technical and require basic development knowledge to be used "as is".

- `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*

3.1.2 Command components

These components are 2nd level layer components desined to implement *user level functionalities*.

- `settings` manage system settings
- `xl2json` process Excel file and store extracted invoice data in JSON format. *More detais here*
- `json2xml` process JSON file and convert it to XML (ROeFact standard compliant)
- `json2pdf` process JSON file and produce invoice as PDF format
- `xml2roefact` upload XML format invoice to "SPV RO eFactura"

3.1.2.1 Session concept

This layer use the concept of **session data**, a session representing "all states & information" for a `Commands` class instance, from its creation until it is destroyed.

The concept allow to use multiple commands (chain commands) in a session, without need of repeating / specifying parameters send to last command process. This is useful to avoid re-requesting end users for parametrs entry in a *web application* or *console session application* (ie, start command and execute multimple commands at a dedicated prompt until a "quit" or similar command).

3.1.2.2 Layer organization

This layer consist of the following clases:

- `CommandResult` is a dataclass aimed to contain resulted information from commands processing. It contains all needed information in order to be able to render and display it as plain text, rich / enenhanced text or HTML. It contains:
  - *status code* (HTTP standard codes)
  - *status short text* as "human representation of status code", *console output* (plain text & HTML)
  - *effective information* resulted from command processing.

- `SessionDataType` is a dataclass containg all potential parameters passed to commands and which are subject to be repeated in the same session (to avoid re-entering them by end users and to present them as proposed default values)
- `Commands` is the final class containing effective methods and session data

For more details see below the *API Reference* section.

3.1.3 Configuration components

These are the components that assure and make possible system configurablitity at user level.

- `config_settings` *USER level* configuration - define application settings & parameters mainly used in invoice info / data detection and extract from invoice Excel format file
- `sys_settings` *SYSTEM level* configuration - system database and parameters, not changeable at user level in current application usage (changing these parameters needs code updating to make them effective) - details in section Sysyem database and parameters

3.1.4 Presentation components

These are 2nd level layer components that make sysyem usable in various forms such as command line console application, daemon / server that runs in background and can be called from local or remote clients, library interfaces (for extensions and custom development) that hide low level technical execution details.

- `app_cli` contains the code for `xl2roefact` application command line (CLI) format
- `__main__` assures right package "addressing" as Python modele (ie, running as `python -m xl2roefact ...`)
- `__version__` keeps current system version and helper functions to assure standard and canonical representation of version string
- `__init__` assure friendly exposing of system public objects (and of course classic pytgon role of "package maker")

3.2 Install library

Library can be installed using 2 methods:

- install from PyPi
- install from site archive of distribution packages

3.2.1 Install from PyPi

The library installation can be done using standard Python instruments:

```
pip install xl2roefact
```

This command will install by default the last stable version. For other versions, standard PyPi procedure to install a specific version must be used.

3.2.2 Install from distribution packages

To install from distribution packages first download the package version intended to install (see download section), choose the package type (if you have no special option, then choose `WHEEL` format) and install it using `pip` as any other Python library installation (*detailed in Python official documentation*).

3.3 Aspecte tehnice referitoare la formatul fisierului JSON aferent facturii

Acest fisier este cel generat de catre aplicatie in urma executiei acesteia cu comanda `xl2json`. Structura de baza a acestui fisier este:

```
{
    "Invoice": {...},
    "meta_info": {...},
    "excel_original_data": {...}
}
```

Cheile de la primul nivel contin:

- **`Invoice`** - datele efective ale facturii
- **`meta_info`**
  - informatii referitoare la procesarea facturii si mapa de conversie a cheii `Invoice` din formatul `JSON` in formatul `XML` cerut de sistemul *RO E-Fact*
  - harta de ajutor in conversia formatului JSON in formatul XML acceptat de sistemul RO E-Fact (cheie `meta_info.map_JSONkeys_XMLtags`) si definititiile XML aferente (cheie `meta_info.invoice_XML_schemes`)
  - alte informatii despre fisierul Excel prelucrat (numele, worksheet cu factura, data si ora procesarii, CRC pentru verificare, etc)
- **`excel_original_data`** - informatiile originale din fisierul Excel, asa cum au fost ele identificate si gasite precum si locatia (adresele celulelor). Aceste informatii sunt utile in cazul in care exista neclaritati in urma procesuluicde conversie pentru "a intelege" de unde si cum arata informatiile originale din fisierul Excel

An example of JSON generated file is available here

3.4 Sysyem database and parameters

System database is an object that interface library components with physical stores of parameters and data requred by system and its applications.

Sometimes it can contain both physical and logical interfaces one example being *InvoiceTypes* which consists of:

- `InvoiceTypes: dict` the physical store of invoice types name and codes
- `InvoiceTypesEnum: Enum` the logical object with invoice types implemented as standard Python enumeration (`enum`)

This let open the possibility that in future versions to "externalize" physical data-objects to other systems or distinct files, but letting small / tinny physical data-objects to stay in `sys_settings.py` module.

3.5 API Refrence (https://invoicetoroefact.renware.eu/xl2roefact/doc/wrapper_810.05a-xl2roefact_DLD_specs.html)

3.6 Download xl2roefact library

- Pachete instalare biblioteca Python formate WHEEL si DIST

4 xl2roefact python library API Reference

version 0.9.dev1

5 libutils

general utilities library for all `xl2roefact` components and modules.

Identification:

- code-name: `libutils`
- copyright: (c) 2023, 2024 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Components:

- `complete_sexe_file() -> bool`: Rename and move resulted exe file (called from `build_sexe` script)

- `dict_sum_by_key(dict, str) -> float`: Sum a dictionary for a given key at all depth levels
- `find_str_in_list(list, list) -> int`: Search more strings (ie, a list) in list of strings
- `hier_get_data_file(file_name: str) -> Path`: Get `Path(file_name)` from hierarchy of locations
- `invoice_taxes_summary(list[dict]) -> dict`: Calculates invoice taxes summary as required by ROefact requirements
- `isnumber(str) -> bool`: Test a string if it could be used as number (int or float)

5.0.0.1 hier_get_data_file

```
def hier_get_data_file(file_name: str) -> Path | None
```

Get `Path(file_name)` from hierarchy of locations: (1) current directory, (2) package `data/` directory, (3) `None` is file does not exists in 1 or 2 locations.

Arguments:

- `file_name` - the name of the file to be returned as full path

Returns:

- `Path` - path of file if was found in (1) or (2) locations or `None` if not found

5.0.0.2 complete_sexe_file

```
def complete_sexe_file(drop_source: bool = True) -> bool
```

Rename and move resulted exe file. This function is dedicated only to development phase, so various objects are hard coded.

Specs:

- `file to process` .../dist_sexe/xl2roefact_to_update_name.exe `-->` .../dist/xl2roefact-version-win64.exe
- Note 1: all function code suppose that current directory is root of `xl2roefact`, ie where is located `pyproject.toml` of package

Arguments:

- `drop_source` - indicate to delete source file after copying, ie make a "move" operation, otherwise make a copy keeping the source file. Default behaviour is to delete source.

Returns:

- `bool` - True if file was found, renamed and moved with no error

5.0.0.3 invoice_taxes_summary

```
def invoice_taxes_summary(invoice_lines: list[dict]) -> list
```

Calculates invoice taxes summary as required by ROefact requirements.

Arguments:

- `invoice_lines` - section with item lines from 'big' invoice dictionary

Returns:

- `list` - usable for "cac_TaxSubtotal" key

5.0.0.4 dict_sum_by_key

```
def dict_sum_by_key(search_dict: dict | list[dict], sum_key: str) -> float
```

Sum all dictionary (or list off dictionaries) items, at all levels, for a given key.

Arguments:

- `search_dict` - dictionary to be searched for
- `sum_key` - key to be searched

Returns:

- `float` - with required sum

5.0.0.5 isnumber

```
def isnumber(a_string: str) -> bool
```

test if a string is valid as any kind of number.

Arguments:

- `a_string` - input string.

Returns:

- `True` - if input string is valid as any kind of number, orherwise `False`.

5.0.0.6 find_str_in_list

```
def find_str_in_list(list_of_str_to_find: list, list_to_search: list) -> int
```

find a substring from `list_of_str_to_find` in elements of `list_to_search`.

Arguments:

- `list_of_str_to_find` - list of strings to search for.
- `list_to_search` - liste where to search for substrings.

Returns:

- `index` - the index of list item which contains `str_to_find` (first found) or `None` if not found.

6 ldxml

ldxml: modul de incarcare a facturii in sistemul ANAF E-Factura

Identification:

- code-name: `ldxml`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XML`
- IESIRI: raport cu validarea si identificatorul incarcarii

7 sys_settings

System database and parameters.

This module acts as an "ORM" between `xl2roefact` system and different data objects. It contains:

- tinny physical data objects (Section 1.)
- logical data objects (Section 2.)
- interfaces to external data objects as files or other specialized systems (Section 2.)

Notes:

- "Sections 1, sl , ..." organization of code even is just a pure visual one, is recommended to be respected and followed it being intended to increase code readability and latter maintainability.
- IMPORTANT to keep in mind: This module IS NOT intended to be modified by end users or administrators. Only development stuff can alter this database because application code must be updated accordingly.
- for updaters remark: because dependencies, code sections should follow strict enumerated order in comments

References:

- copyright: (c) 2024 RENWare Software Systems

- author: Petre Iordanescu (petre.iordanescu@gmail.com)

7.0.0.1 InvoiceTypes

Section 2. INTERFACES & LOGICAL data

8 rdinv

rdinv: modul de procesare a fisierului Excel ce contine factura si colectare a datelor aferente.

Formatul acceptat fisier Excel este `XLSX`.

Identification:

- code-name: `rdinv`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier format XLSX ce contine factura emisa (cod: `f-XLSX`)
- IESIRI: fisier format JSON imagine a datelor facturii (cod: `f-JSON`)

8.0.0.1 rdinv

```
def rdinv(file_to_process: str,
          invoice_worksheet_name: str = None,
          *,
          invoice_type_code: InvoiceTypesEnum = InvoiceTypesEnum.NORMALA,
          debug_info: list[str] = None,
          owner_datafile: Path = None) -> dict
```

read Excel file for invoice data.

Produce a dictionary structure + JSON file with all data regarding read invoice: canonical KV data, meta data, map to convert to XML and original Excel data.

Arguments:

- `file_to_process` - the invoice file (exact file with path).
- `invoice_worksheet_name` - the worksheet containing invoice, optional, defaults to first found worksheet.
- `invoice_type_code` - code of invoice type, for example "380" for regular.
- `debug_info` - list with *index[0]* containing all print messages issued by function. If None or nothing sent will print on `stdout`. List is required because a mutable object is needed to be able to write in.

- `owner_datafile` - specify a file to read supplier data from, default `None` meaning to read supplier data from Excel file.

Returns:

- `dict` - the invoice extracted information from Excel file as `dict(Invoice: dict, meta_info: dict, excel_original_data: dict)`

Notes:

- `db: pylightxl object`: EXCEL object with invoice (as a whole)
- `ws: pylightxl object`: WORKSHEET object with invoice

8.0.0.2 get_excel_data_at_label

```
def get_excel_data_at_label(pattern_to_search_for: list[str],
                            worksheet: xl.Database.ws,
                            area_to_scan: list[list[int]] = None,
                            targeted_type: Callable = str,
                            down_search_try: bool = True) -> dict
```

get "one key Excel values", like invoice number or invoice issue date.

Arguments:

- `pattern_to_search_for` - for example for inv number, will pass the `PATTERN_FOR_INVOICE_NUMBER_LABEL`.
- `worksheet` - the worksheet containing invoice (as object of `pyxllight` library).
- `area_to_scan` - area of cells to be searched, default whole worksheet.
- `targeted_type` - what type expect (will try to convert to, if cannot will return str), default `str`.
- `down_search_try` - establish if DOWN search method is tried, default `True`.

Returns:

`None` if not found OR `dictionary` containing: * `"value": int | float | str` - the value found covenrted to requested `targeted_type` if possible or `str` otherwise; if "out of space" then returns `None` * `"location": (row, col)` - adrees of cell where found value

Notes:

- normal scan order is 1.RIGHT, 2.DOWN (if allowed), 3.IN-LABEL only in given area and pattern.

8.0.0.3 mk_kv_invoice_items_area

```
def mk_kv_invoice_items_area(invoice_items_area_xl_format) -> dict
```

transform `invoice_items_area` in "canonical JSON format" (as kv pairs).

Arguments:

- `invoice_items_area_xl_format` - invoice items area in Excel format (ie, DataFrame with row, col, data).

Returns:

- `invoice_items_area_xl_format` - dictionary with invoice items in Excel format (ie, rows, columns).

Notes:

- for ROefact XML model (& plan) see `invoice_files/__model_test_factura_generat_anaf.xml`.

8.0.0.4 get_invoice_items_area

```
def get_invoice_items_area(worksheet, invoice_items_area_marker,
                           wks_name) -> dict
```

get invoice for `invoice_items_area`, process it and return its Excel format.

Process steps & notes:

- find invoice items subtable.
- clean invoice items subtable.
- extract relevenat data.
- NOTE: all Excel cell addresses are in `(row, col)` format (ie, Not Excel format like "A:26, C:42, ...")

Arguments:

- `worksheet` - the worksheet containing invoice (as object of `pyxllight` library).
- `invoice_items_area_marker` - string with exact marker of invoice items table.
- `NOTE` - this is the UPPER-LEFT corner and is determined before calling this procedure.
- `wks_name` - the wroksheet name (string) of the `worksheet` object.

Returns:

- `invoice_items_area` - dictionary with invoice items in Excel format (ie, rows, columns).

8.0.0.5 get_merged_cells_tobe_changed

```
def get_merged_cells_tobe_changed(file_to_scan,
                                  invoice_worksheet_name,
                                  keep_cells_of_items_ssd_marker=None) -> list
```

scan Excel file to detect all merged ranges.

Arguments:

- `file_to_scan` - the excel file to be scanned.
- `invoice_worksheet_name` - the worksheet to be scanned.

- `keep_cells_of_items_ssd_marker` - tuple with cells that will be marked IN ANY CASE to be preserved:
- use case: to keep all potential invoice items ssd rows.
- format: `tuple(row, col, val)` where row & col are relevant here
- default: `None`

Returns:

- `cells_to_be_changed` - list with cells that need to be chaged in format `(row,col)`.

Notes:

- function is intended to be used ONLY internal in this module.
- use `openpyxl` library to do its job.

8.0.0.6 build_meta_info_key

```
def build_meta_info_key(excel_file_to_process: str,
                        invoice_worksheet_name: str, ws_size: list,
                        keyword_for_items_table_marker: str,
                        found_cell: list) -> dict
```

build meta_info key to preserve processed Excel file meta information: start address, size.

Notes:

- (1.) all cell addresses are in format (row, col) and are absolute (ie, valid for whole Excel file).
- (2.) this function is designed to be used internally by current module (using outside it is not guaranteed for information 'quality').

Arguments:

- `excel_file_to_process` - name of file to process as would appear in `meta_info` key.
- `invoice_worksheet_name` - the worksheet name as would appear in `meta_info` key.
- `ws_size` - worksheet size as would appear in `meta_info` key (index 0 max rows, index 1 max columns).
- `keyword_for_items_table_marker` - the content of cell used as start of invoice items subtable as would appear in `meta_info`.
- `found_cell` - position of cell used as start of invoice items subtable as would appear in `meta_info` key (index 0 row, index 1 column).

Returns:

- `meta_info` - dictionary built with meta information to be incorpoarted in final invoice dict

8.0.0.7 get_partner_data

```
def get_partner_data(partner_type: str,
                     *,
                     wks,
                     param_invoice_header_area: dict,
                     supplier_datafile: Path = None) -> None
```

Get invoice partener data from Excel.

Notes:

- *for developers*: function works by generating side effects and must be located in `rdinv.py`
- *side effects*: this function works by directly modifying `param_invoice_header_area` sent parameter
- *supplier_datafile exception*: if file is not found or cannot be read, this function will force complete application termination (`sys.exit`)

Arguments:

- `partner_type` - one of "CUSTOMER", "SUPPLIER" or "OWNER" to specify for what kind of parner get data. The value "OWNER" is designed to get data from an outside database / file (master data)
- `wks` - current work-on `pylightxl Worksheet` object
- `param_invoice_header_area` - *mode IN-OUT*, outside `param_invoice_header_area` as used and needed in `rdinv()`. This function will write back in this variable
- `supplier_datafile` - for `partner_type = "CUSTOMER"` here is expected the file where to get supplier data

Returns:

- `None` - all data is produced directly in parameters as side effect

9 chkisld

chkisld: modul de verificare a starii de incarcare a unei facturi emise

Identification:

- code-name: `chkisld`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XLSX` sau numarul / cheia / codul facturii

- IESIRI: valoarea echivalent `TRUE` daca factura a fost deja incarcata sau valoare echivalent `FALSE` daca factura nu a fost incarcata

10 __init__

10.0.0.1 __version__

default conversion takes place over xl2roefact actual version

11 wrxml

wrxml: modul de generare a fisierului format XML

Identification:

- code-name: `wrxml`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-JSON`
- IESIRI: fisier format XML conform cerintelor si sistemului ANAF E-Factura (cod: `f-XML`)

12 __main__

xl2roefact.main: Python package standard file to assure run as `python -m xl2roefact`.

Identification:

- code-name: `__main__`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Deployments:

- Windows: MSI installer with EXE application.
- Linux: `xl2roefact` executable shell as wrapper for `xl2roefact.py`.

Specifications:

- command general format: `python -m xl2roefact [OPTIONS] COMMAND [ARGS]...`.

- help: `python -m xl2roefact --help`.

13 config_settings

Configuration and setting parameters.

Regulile recomandate se gasessc in documentul (recommended rules are in document `xl2roefact/data/README_app_config_rules.md`)

Public objects:

- `rules_content`: contains the rules text (rendered)

Info:

- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

13.0.0.1 DEFAULT_DUE_DATE_DAYS

NOTE: "pattern-uri" (sabloane) de identificare si regasire a datelor folositi de

__ comanda `xl2json` reprezentind functionalitatea de extragere a datelor din Excel si exportul lor in formatul JSON (modulul `rdinv)__

13.0.0.2 rules_content

`rules_content` public variable to be use as "mini help" by `settings -r` command of application

13.0.0.3 load

```
def load()
```

Read and load settings from external data file.

14 commands

Layer 2 commands API implementation.

Objectives:

- create an environment where a xl2roefact can be run in *session or interactivelly mode*
- session parameters: persist commands run parameters in user profile (directory of `os.%userprofile%` or Linux `~/.profile`)

- group all layer 2 commands for:
  - `xl2roefactd` (aka server)
  - `xl2roefact-client` (aka console client)
  - `web2roefact` (aka web client UI front end)* components

Identification:

- code-name: `commands`
- Copyright: (c) 2024 RENware Software Systema
- Author: Petre Iordanescu (petre.iordanescu@gmail.com)

14.1 CommandResult Objects

```
@dataclass
class CommandResult()
```

Define the result of execution. This structure contains information collected in methods execution. After each method execution all "prints" and status information stated by method in its execution (ie, which was saved) will be contained in.

Fields:

- `status_code`: `int` Status code as used by HTTP standard returned codes (200 for success)
- `status_timestamp`: `str` Timestamp of information in UTC ISO format
- `status_text`: `str` Short text of this result set. Normally used to display a brief message note associated to code (for example "404 Not found")
- `result`: `Any` The effective result information returned by method as core result of execution. Depending on method, this is a *Python specific structure*, scalar, basic or complex one
- `stdout_text`: `str` Collected console "prints" output in text format. Normally a standard print() of this value will reproduce the exact console output if method would be "raw executed" in development mode
- `stdout_html`: `str` the same as `stdout_text` but in HTML format ready to be sent "as is" to a browser (its a COMPLETE and FULL HTML doc). Used if pages together with other elements it is recommended to isolate it with distinct `div` and `iframe` tags

14.2 SessionDataType Objects

```
@dataclass
class SessionDataType()
```

Define session data used in class `Commands`. These data objects mainly represents almost all parameters encountered in console application and that are suspect to be reusable when "chain more commands" iin the same session and is desirable to keep last values.

Also in web applications is normal to not ask for parameters already entered by an end user and to preserve last entered values at least as default ones.

14.3 Commands Objects

```
class Commands()
```

xl2roefact commands layer implementation. Descriere generala layer si componenta..

14.3.0.1 __init__

```
def __init__()
```

Init session data variables with default values.

14.3.0.2 session_data_set

```
def session_data_set(*,
                     file_name: str = ...,
                     invoice_type: InvoiceTypesEnum = ...,
                     files_directory: Path = ...,
                     owner_datafile: Path = ...,
                     verbose: bool = ...) -> bool
```

Set session data.

Rules:

- session data is kept as class-instance variables. This will use for "interactive" or "web" editions
- if a parameter is not sent at call, then it is left unchanged
- any other sent value is saved as instance variable
- elipsis as default parametrs values help to make difference between a sent parameter (even with None) and a not sent one

Arguments:

- `all_item` - more instances = all data items required to be kept as reusable session data

Returns:

- `bool` - any change was made

14.3.0.3 session_data_reset

```
def session_data_reset()
```

Resset session data to defaults.

14.3.0.4 version

```
@classmethod
def version(cls) -> str
```

return the version of `xl2roefact` used by this class

14.3.0.5 xl2json

```
def xl2json(invoice_type: InvoiceTypesEnum = ...,
            file_name: str = ...,
            files_directory: Path = ...,
            owner_datafile: Path = ...,
            verbose: bool = ...) -> bool
```

read excel invoice and generate a JSON file with invoice data, miscellaneous meta and original Excel found data

Arguments:

- `invoice_type_code` - invoice type (for example regular invoice or storno) as this info is not usually subject of Excel file. Default to `380` (regular / usual invoice)
- `file_name` - files to process (wildcards allowed).
- `files_directory` - directory to be used to look for Excel files. Defaults to `invoice_files/`. NOTE: if default directory does not exists will consider current directory instead
- `owner_datafile` - File to read invoice supplier (owner) data instead Excel.
- `verbose` - show detailed processing messages". Defaults to `False`.

Returns:

- `bool` - True if command executed without errors. If return in False, the kasr result should be inspected to see error status and text (method `results_stack_pop()`)

14.3.0.6 response_out

```
def response_out(*,
                 status_code=200,
                 status_text="undefined",
                 status_result="undefined")
```

Prepare and enque a response. This is designed to be in-class used and not for public interface. Arguments are min of what to be enqueued. Other ibformation are constructed local.

14.3.0.7 get_last_result

```
def get_last_result() -> dict[CommandResult]
```

Get last result dictionary from stack WITHOUT drooping it.

Returns:

- `CommandResult` - last result as dictionary

14.3.0.8 pop_session_results

```
def pop_session_results() -> list[CommandResult]
```

Get all session results as dictionary.

Returns:

- `CommandResult` - list with all session results as dictionary

14.3.0.9 get_var_name

```
@classmethod
def get_var_name(cls, var)
```

Return a variable defined in class as string of its name.

15 app_cli

app_cli: the command line application for all xl2roefact functionalities.

Identification:

- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

15.0.0.1 about

```
@app_cli.command()
def about()
```

Provide a short application description.

15.0.0.2 settings

```
@app_cli.command()
def settings(rules: Annotated[
    bool,
    typer.
    Option("--rules", "-r", help="show settings recommended update rules"),
] = False)
```

Display application configuration parameters and settings that are subject to be changed by user.

Arguments:

- `rules` - show recommended rules to follow when change application configurable settings (available in both RO & EN languages). Defaults to `False`.

15.0.0.3 xl2json

```
@app_cli.command()
def xl2json(
    invoice_type: InvoiceTypesEnum = InvoiceTypesEnum.NORMALA.value,
    file_name: Annotated[
        str, typer.Argument(
            help="files to process (wildcards allowed)")] = "*.xlsx",
    files_directory: Annotated[
        Path,
        typer.Option(
            "--files-directory",
            "-d",
            exists=False,
            file_okay=False,
            dir_okay=True,
            writable=True,
            readable=True,
            resolve_path=True,
            help=
            "directory to be used to look for Excel files (if default directory does not exists
will consider current directory instead)."
        ),
    ] = "invoice_files/",
    owner_datafile: Annotated[
        Path,
        typer.Option(
            "--owner-datafile",
            "-o",
            exists=False,
            file_okay=True,
            dir_okay=False,
            writable=False,
            readable=True,
            resolve_path=False,
            help="File to read invoice supplier (owner) data instead Excel."),
    ] = None,
    verbose: Annotated[
        bool,
        typer.
        Option("--verbose", "-v", help="show detailed processing messages"),
    ] = False)
```

Extract data from an Excel file (save data to JSON format file with the same name as original file but `.json` extension).

Arguments:

- `invoice_type_code` - invoice type (for exaple regular invoice or storno) as this info is not usually subject of Excel file. Default to `380` (regular / usual invoice)
- `file_name` - files to process (wildcards allowed).
- `files_directory` - directory to be used to look for Excel files. Defaults to `invoice_files/`. NOTE: if default directory does not exists will consider current directory instead
- `owner_datafile` - File to read invoice supplier (owner) data instead Excel.
- `verbose` - show detailed processing messages". Defaults to `False`.

15.0.0.4 called_when_no_command

```
@app_cli.callback(invoke_without_command=True)
def called_when_no_command(
    ctx: typer.Context,
    version: Annotated[
        bool,
        typer.Option("--version", "-V", help="show application version"),
    ] = False)
```

Application global information (command agnostic).

15.0.0.5 run

NOTE: for `run` "reason to be" as copy of `app_cli` see iss `0.1.22b 240216piu_a`

16 chkxml

chkxml: modul de validare a facturii in sistemul ANAF E-Factura

Identification:

- code-name: `chkxml`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XML`
- IESIRI: raport cu eventualele erori de validare

17 _tst_dropme2

18 __version__

`xl2roefact` version info.

```
######## ####             ########  ######
#  ##  # #  #             #      # ##    ##
##    ## #  #             ###  ### #  ##  #
##  ##  #  #               #  #   #  ##  #
##    ## #  #                #  #   #  ##  #
#  ##  # #  #####           #  #   #  ##  #
#  ##  # #      #           #  #   ##    ##
######## ########           ####    ######

                          ######## ########  ######   ######  ########
#######   ######          #      # #     #  ##    ## ##    ## #      #
#     ## ##    ## #######  #  ##### #  ####  #  ##  # #  ##  # ###  ###
#  ##  # #  ##  # #     #  #    #   #    #   #      # #  #####   #  #
#  ##### #  ##  # #######  #  ###   #  ###   #  ##  # #  #####   #  #
#  # #   #  ##  #          #  ##### #  #     #  ##  # #  ##  #   #  #
#  #  #  ##    ##          #      # #  #     #  ##  # ##    ##   #  #
####   #  ######           ######## ####    ########  ######    ####
```

18.0.0.1 __version__

current 0.9, previous 0.8

18.0.0.2 normalized_version

```
def normalized_version(raw_version: str = __version__) -> str
```

transform version string in canonical form.

Used in `__init__.py` to return `__version__` object as will be seen by package consumers

Arguments:

- `raw_version` - a raw version string. Defaults to package current version string.

Returns:

`str:` canonical version string

19 data

xl2roefact in-package designed to implement **data layer**

- Issued: 2024 March
- Author: Petre Iordanescu, petre.iordanescu@gmail.com
- (c) RENware Software Systems

II.III aplicatie desktop

II.IV aplicatie mobile

RENware Software Systems

20 xl2roefact - resurse pentru descarcare (downloads)

20.1 Formate sursa biblioteca Python

20.1.1 xl2roefact din PyPi

Versiunea de pe pe repository-ul public `PyPi` permite instalarea directa in mediul Python local astfel:

```
pip install xl2roefact
```

In acest mod va fi instalata automat *ultima versiune stabila* publicata pe *PyPi (https://pypi.org/project/xl2roefact/)* . Accesati linkul anterior pentru a putea accesa alte versiuni publicate pe *PyPi* si modul de instalare a acestora.

20.1.2 xl2roefact pachete sursa versiuni anterioare

0.8

- `0.8` BUGFIX xl2roefact entry point (https://pypi.org/project/xl2roefact/0.8/)

0.7

- `0.7` clean xl2roefact package and invoice JSON (https://pypi.org/project/xl2roefact/0.7/)
- `0.7rc2` updated console application to run in silent or vebosed (https://pypi.org/project/xl2roefact/0.7rc2/)
- `0.7rc1` review & clean `xl2roefact.rdinv` module of TODOs (https://pypi.org/project/xl2roefact/0.7rc1/)
- `0.7rc0` settings refresh option and app settings reload by request (https://pypi.org/project/xl2roefact/0.7rc0/)

0.6

- `0.6` invoice type JSON-XML tag (https://pypi.org/project/xl2roefact/0.6)
- `0.6rc0` system database and parameters (https://pypi.org/project/xl2roefact/0.6rc0/)

20.2 Formate pentru Windows x64

20.2.1 xl2roefact linie comanda kit instalare (win64-msi)

0.8

- `0.8` BUGFIX xl2roefact entry point - INDISPONIBILA

0.7

- `0.7` clean xl2roefact package and invoice JSON
- `0.7rc2` updated console application to run in silent or vebosed
- `0.7rc1` review & clean `xl2roefact.rdinv` module of TODOs - INDISPONIBILA
- `0.7rc0` settings refresh option and app settings reload by request

0.6

- `0.6` invoice type JSON-XML tag
- `0.6rc0 system database and parameters` - INDISPONIBILA

20.2.2 xl2roefact linie comanda executabil portabil (win64-exe)

0.8

- `0.8` BUGFIX xl2roefact entry point - INDISPONIBILA

0.7

- `0.7` clean xl2roefact package and invoice JSON
- `0.7rc2` updated console application to run in silent or vebosed
- `0.7rc1` review & clean `xl2roefact.rdinv` module of TODOs - INDISPONIBILA
- `0.7rc0` settings refresh option and app settings reload by request

0.6

- `0.6` invoice type JSON-XML tag
- `0.6rc0 system database and parameters` INDISPONIBILA

20.3 Sablon fisier configurare a aplicatiei xl2roefact

Sablonul permite configurarea aplicatiei prin modificarea fragmentelor de text care trebuiesc cautate in fisierul Excel pentru identificarea diverselor informatii aferente facturii.

Sablonul este in format YAML (https://yaml.org/) iar informatiile ce trebuiesc descrise sunt explicate individual in comentarii insotitoare. De asemenea este util a fi citite si recomandarile date in pagina de descriere a aplicatiei.

Pentru a beneficia de cobfigurarile facute de dumneavoastra trebuie sa creati un fisier `app_settings.yml` in directorul curent din care lansati aplicatia, fisier ce contine noile configurari dorite. **Numele fisierelui este obligatoriu a fi respectat.**

Fisiere de configurare multiple

De retinut ca acest fisier este considerat (daca exista) cel din directorul curent de unde lansati aplicatia. Deci daca v-ati creat mai multe directoare de lucru (de exemplu pentru clienti diferiti) puteti crea fisiere de configurare specifice, cite unul in fiecare director.

- Descarcare sablon de fisier de configurare

Continutul sablonolui

De mentionat ca acest sablon este pre-completat cu situatii curent intilnite in practica, el fiind chiar sablonul implicit folosit de aplicatie.

20.4 Sablon fisier cu date furnizor

- Descarcare fisier cu date furnizor

III. web2roefact

21 Under construction page

UPCOMING...

IV. invoice template

22 invoice template

22.1 Instalarea sablonului de factura emisa

Aceasta componenta consta dintr-un director (ce nu necesita instalare speciala ci simpla copiere locala acolo unde va fi utilizat). Acest director contine:

- fisierul pentru factura `invoice_template_CU_tva.xlsx` ce este disponibil pentru descarcare aici
- directorul `released_packages/` ce contine versiuni anterioare de sablon ce sunt inca suportate
- prezentul document

22.2 Recomandari in utilizarea sablonului

Aceasta sectiune se refera la modul in care ar trebui "tratat" continutul fisierului Excel cu factura ***in conditiile in care se intentioneaza ca aceasta sa fi procesata ulterior cu sistemul*** `INVOICEtoROeFact`. Acest sablon este general valabil (este un fisier Excel ca oricare altul) deci in acest caz este important a "constientiza" faptul ca informatia aferenta facturii din Excel va fi cautata, identificata si gasita in scopul de a fi salvata in formatele de factura electronica (utilizarea acestui sablon de factura Excel impreuna cu sistemul `INVOICEtoROeFact` **ESTE O VARIANTA DE LUCRU RECOMANDATA**, dar nu obligatorie).

Astfel **se recomanda ca acest sablon sa fie utilizat asa cum este livrat**, fara a efectua modificari majore in structura sa cum ar fi:

- modificarea formatelor (de tip de date) celulelor in scopul unei afisari "mai frumoase"
- adaugarea de informatii prin concatenare de siruri de caractere sau orice alte metode de a altera continutul vizibil al celulelor in scopul unei afisari "mai frumoase"
- modificarea locatiilor celulelor prin inserarea sau stergerea de linii, coloane sau celule noi

> **In general trebuie facuta diferenta intre datele facturii si modul in care aceasta va fi tiparita (va aparea la tiparire / previzualizare).**

DETALII TEHNICE:

Continutul informational al facturii nu trebuie nici confundat si nici mixat cu **formatul de afisare al acesteia** (layout). Pentru acesta din urma se recomanda a fi folosite cu precadere *regulile de formatare* din Excel si nu cele stocare a datelor. Un exemplu este un numar zecimal oarecare unde:

- una este valoarea introdusa intr-o celula (de ex cu 3 zecimale) si
- alta este valoarea afisata (cu 2 zecimale) - aceasta din urma trebuie obtinuta prin formatarea celulei respective de a afisa 2 zecimale prin rotunjire insa valoarea efectiva trebuie sa fie cea originala cu 3 zecimale, lucru (diferenta) care se poate vedea la editarea continutului celulei.

22.3 Licenta

RENware Software Systems

23 Sablon template factura. Resurse pentru descarcare

Format XLSX Office Excel

- `0.1.20` office Excel XLSX
- `0.1.20` arhiva ZIP
- `0.1.11` arhiva ZIP

V. Help

V.I Manuale web2roefact

INVOICEtoROefact System

24 Catalogul manualelor de utilizare

Cuprins:

INVOICEtoROefact System

25 Catalogul manualelor de configurare si administrare

Cuprins:

V.II Cerintele sistemului

RENware Software Systems

26 Propunere tehnica

Cuprins:

- Client: n/a - not public
- Data: 2023-Noiembrie

Codificarea documentelor

- codificarea numelor documentelor si a proceselor este facuta in conformitate cu metodologia *RENware SDEVEN (http://sdeven.renware.eu)*

Cuprins:

- Propunere tehnica
  - Aria de cuprindere
  - Considerente generale de securitate
  - Considerente generale privind bazele de date proprii sistemelor
  - Considerente generale privind auditarea informatiilor

26.1 Aria de cuprindere

Solutiile propuse prin aceasta propunere tehnica sunt:

- **INVOICEtoROefact** (*code-name* `api_to_roefact`) integrare Sistemul National de Facturi Emise *RO e-Factura* descriere si cerinte aici
- **PayValidaBoa** (*code-name* `payments_validation_board`) Flux aprobare facturi primite pentru ordonantare la plata descriere si cerinte aici

In continuare se prezinta o serie de considerente generale valabile pentru toate sistemele din aria de acoperire.

26.2 Considerente generale de securitate

- **(RSEC-01)** fisierele de configurare a sistemelor (fiind format text `UTF-8`) vor avea ca `owner` un utilizator dedicat sistemului respectiv sau utilizatorul `root`. Numai acesti doi utilizatori pot avea acces `RW` la aceste fisiere
- **(RSEC-02)** toate documentele de provenienta externa sistemelor vor fi "purtatoare" ale unui certificat digital ce **atesta validitatea documentelor**. Acest certificat va fi de preferinta de tip *"semnatura electronica"* dar nu obligatoriu calificata. Este suficient un simplu certificat (cheie) tip `RSA` generat intern si distribuit utilizatorilor autorizati sa emita documentele respective. O copie a certificatului (sau a certificatelor daca se vor emite mai multe) ce atesta validitatea unui document va sta pe server in locatii ce sunt conforme cu *RSEC-01*

26.3 Considerente generale privind bazele de date proprii sistemelor

- **(DBS-01)** bazele de date vor contine o cheie primara *"real primara"* (adica avind toate caracteristicile tehnice pentru `PK` in sensul uzual cunoscut din teoria bazalor de date). Aceasta cheie va fi de tip `Char(32)` reprezentind tipul `uuid4` (cunoscut si ca `guid`) convertit la sir de caractere `UTF-8` si reversibil ca transformare din `string` in `uuid4`. Aceasta cheie va fi generata automat si intretinuta de sistem deservind scopuri pur tehnice de *referentiere si relationare* a datelor. Modificarea manuala nu este permisa putind genera situatii de hazard.
- **(DBS-02)** bazele de date vor contine si o alta *"cheie primara uman recongnoscibila"* (`AK` in teoria bazelor de date) utilizata in scop de **recunoastere si regasire** a informatiei de catre utilizatori. Aceasta cheie va avea urmatoarele catacterisrici:
  - va fi *unica*, tip `Char(10)` (limitarea lungimii se va aplica la introducerea datelo si nu in baza de date)
  - *agnostic case*, nu se va face diferenta intre litere mari sau mici (pentru a evita confuziile)
  - *obligatorie* iar daca utilizatorul "nu o doreste" se va default-a la `PK-ul` anterior
- **(DBS-03)** bazele de date vor fi intr-unul din formatele: **(a) relational** sau **(b) JSON standard**. Pentru bazele de date in format relational va fi preferata o solutie de SGBD tip open source matura, intretinuta in urmatoarea ordine de aplicare:
  - `1.` *SQLite (https://www.sqlite.org/index.html)* pentru baze de date ce nu vor depasi 10,000 de inregistrari
  - `2.` *PostgreSQL (https://www.postgresql.org/)* pentru baze de date ce se esttimeaza ca vor depasi 10,000 de inregistrari
  - `3.` *MariaDB (https://mariadb.org/)* pentru baze de date ce se esttimeaza ca vor depasi 10,000 de inregistrari
  - prima varianta va fi preferata datoritra "portabilitatii datelor"
  - a treia varianta este enumerata ca optiune preferata a utilizatorului la varianta `2.`

- **(DBS-04)** bazele de date vor folosi numai cimpuri formate standard, clasice si elemetare:
  - sir de carectere (`CHAR` sau `VARCHAR`)
  - numere intregi cu semn (`INTEGER`)
  - numere reale cu semn (`FLOAT`)
  - numere combinate a caror valoare poate fi intreg sau real (`NUMBER`)
  - valori logice sub forma intreg cu semn astfel: `1` pentru TRUE si `0` sau `NULL` pentru FALSE
  - valori logice sub forma de caracter astfel: prima litera din lista `[Y, y, D, d, T, t]` pentru TRUE si orice altceva inclusiv `NULL` pentru FALSE
- **(DBS-05)** in cazul bazelor de date relationale, integritatile referentiale vor fi evitate la maximum prin intretinerea datelor numai cu ajutorul aplicatiei sau in cazull necesitatii modificarii manuale a datelor, aceasta modfica re sa fie efectuata numai de personal calificat
- **(DBS-06)** informatiile de tip data-timp (data, ora, etc...) vor fi stocate de preferinta sub forma de `String` in formatul ISO: `YYYY-MM-DD HH:MM:SS.nnnnn`.
- **(DBS-07)** informatii de data-timp vor fi stocate avind valori agnostice de "Time Zone" adica vor fi considerate `UTC` lucru care va permite comparabilitatea acestora indiferent de locatia /zpna de timp de unde au fost generate.

26.4 Considerente generale privind auditarea informatiilor

- Cimpurile de audit ce indica utilizatori:
  - **(AUD-01)** pentru informatiile CONSTIENT GENERATE DE UTILIZATORI (adica generate prin activarea unor controale vizuale, prin lansarea manuala a unei aplicatii, etc), aceste cimpuri vor contine **numele tip `username` al utilizatorului folosit pentru autentificarea in sistem**
  - **(AUD-02)** pentru informatiile GENERATE DE SISTEM la rulari automate, periodice, de verificare, de validare, etc, aceste cimpuri vor contine textul **`system`** (pentru a evita confuzii cu utilizatori reali la nivel de sistem de operare)
- **(AUD-03)** Cimpurile de audit ce indica date calendaristice vor respecta standardul ISO fiind in formatul maximal `YYYY-MM-DD hh:mm:ss`

INVOICEtoROefact System

27 Propunere tehnica sistem INVOICEtoROefact

Cuprins:

- p/n: `0000-0095`
- code-name: `api_to_roefact`
- commercial name: **INVOICEtoROefact**
- url propunere tehnica: `http://apitoroefact.renware.eu/commercial_agreement/110-SRE-api_to_roefact_requirements.html`
- git: `https://github.com/petre-renware/api_to_roefact`

Cuprins:

- Propunere tehnica sistem INVOICEtoROefact
  - Obiective
  - Vedere de ansamblu a solutiei
  - Componenta xl2roefact
    - Diagrama logica de functionare a componentei
  - Componenta WEB_DASHB
  - Componenta SYSTEM_DB

27.1 Obiective

Acest sistem va asigura incarcarea facturilor emise in sistemul ANAF E-Factura (https://www.anaf.ro/anaf/internet/ANAF/despre_anaf/strategii_anaf/proiecte_digitalizare/e.factura) cu respectarea reglementarilor publicate in acest sens (lista contine si legaturi catre fisierele publicate de catre ANAF):

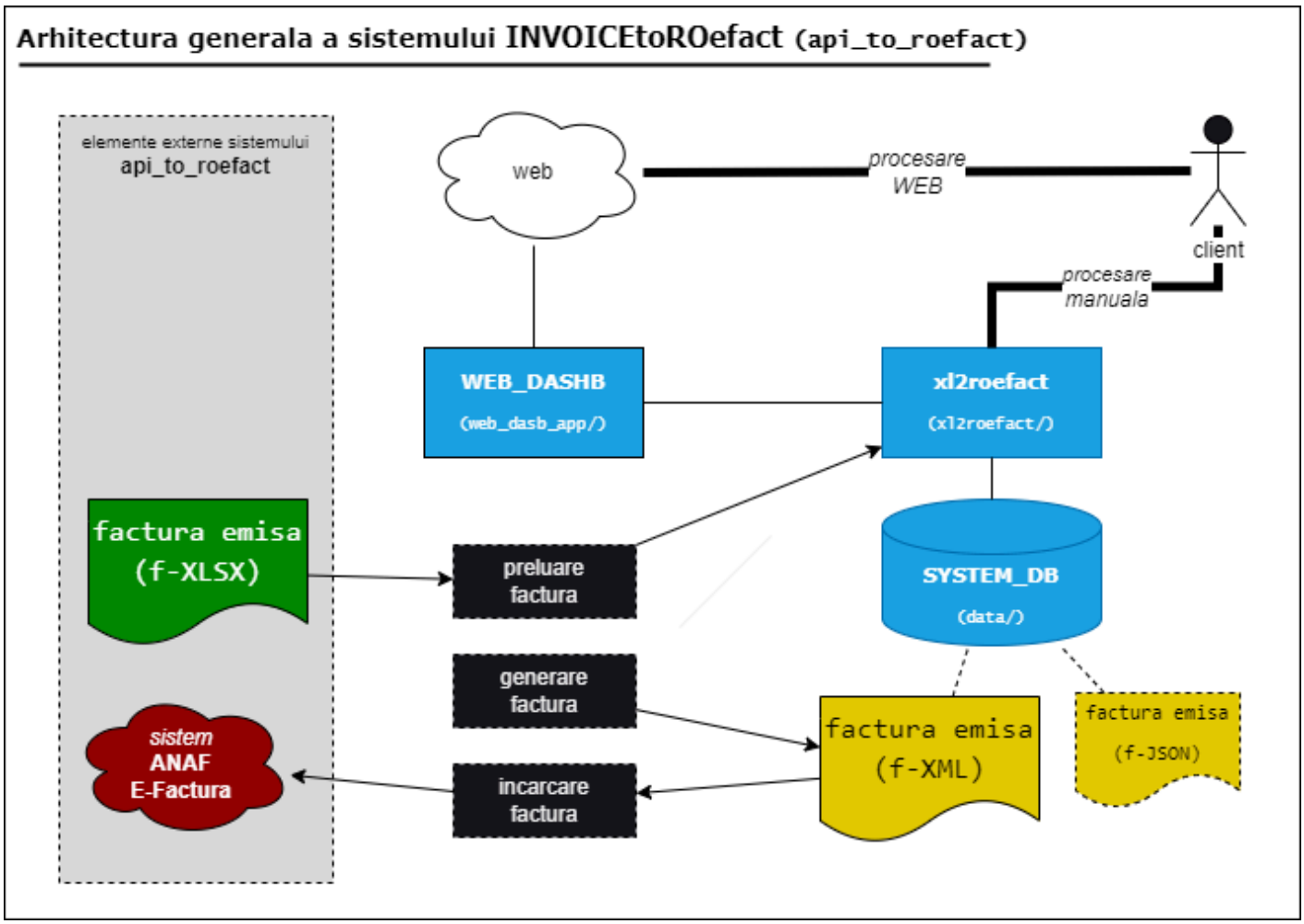
- Conformarea la modificarile legislative si utilizarea sistemului (https://static.anaf.ro/static/10/Anaf/Informatii_R/Informatii_modificare_CIUS_RO.pdf)
- Informatii de interes referitoare la implementarea sistemului național privind factura electronică RO e-Factura (https://static.anaf.ro/static/10/Anaf/Informatii_R/Comunicat_e-factura_aprilie2022_v2_050422.pdf)
- Instrucțiuni de utilizare (https://static.anaf.ro/static/10/Anaf/Informatii_R/API/Oauth_procedura_inregistrare_aplicatii_portal_ANAF.pdf)

27.2 Vedere de ansamblu a solutiei

Solutia `api_to_roefact` consta din urmatoarele componente:

- `api_to_roefact`.**`xl2roefact`** aceasta componenta are rolul de a implementa efectiv obiectivele principale ale sistemului **INVOICEtoROefact**. Componenta este capabila sa ruleze atit "*standalone*" (ca linie de comanda CLI) dar si prin utilizarea ei de catre componenta `WEB_DASHB` si astfel utilizarea ei in varianta de sistem prezentat "*over internet / intranet*". Prezentarea detalita a acesteia se gaseste aici.
- `api_to_roefact`.**`WEB_DASHB`** aceasta componenta are rolul de agrega componentele si de a prezenta solutia **INVOICEtoROefact** "*over internet / intranet*". De asemenea componenta asigura modulele UI necesare pentru administrarea sistemului. Prezentarea detalita a acesteia se gaseste aici.
- `api_to_roefact`.**`SYSTEM_DB`** .Aceasta componente reprezita baza de date a sistemului **INVOICEtoROefact** atit partea relationala dar si partea `no-sql` a acesteia (utilizata pentru eventuale sincrnizari provenite din utilizarea CLI a componentei `xl2roefact`). Prezentarea detalita a acesteia se gaseste aici.

Figura urmatoare prezinta schematic rolul general al componentelor precum si interactiunea acestora cu mediul exterior sistemului `api_to_roefact`.



27.3 Componenta xl2roefact

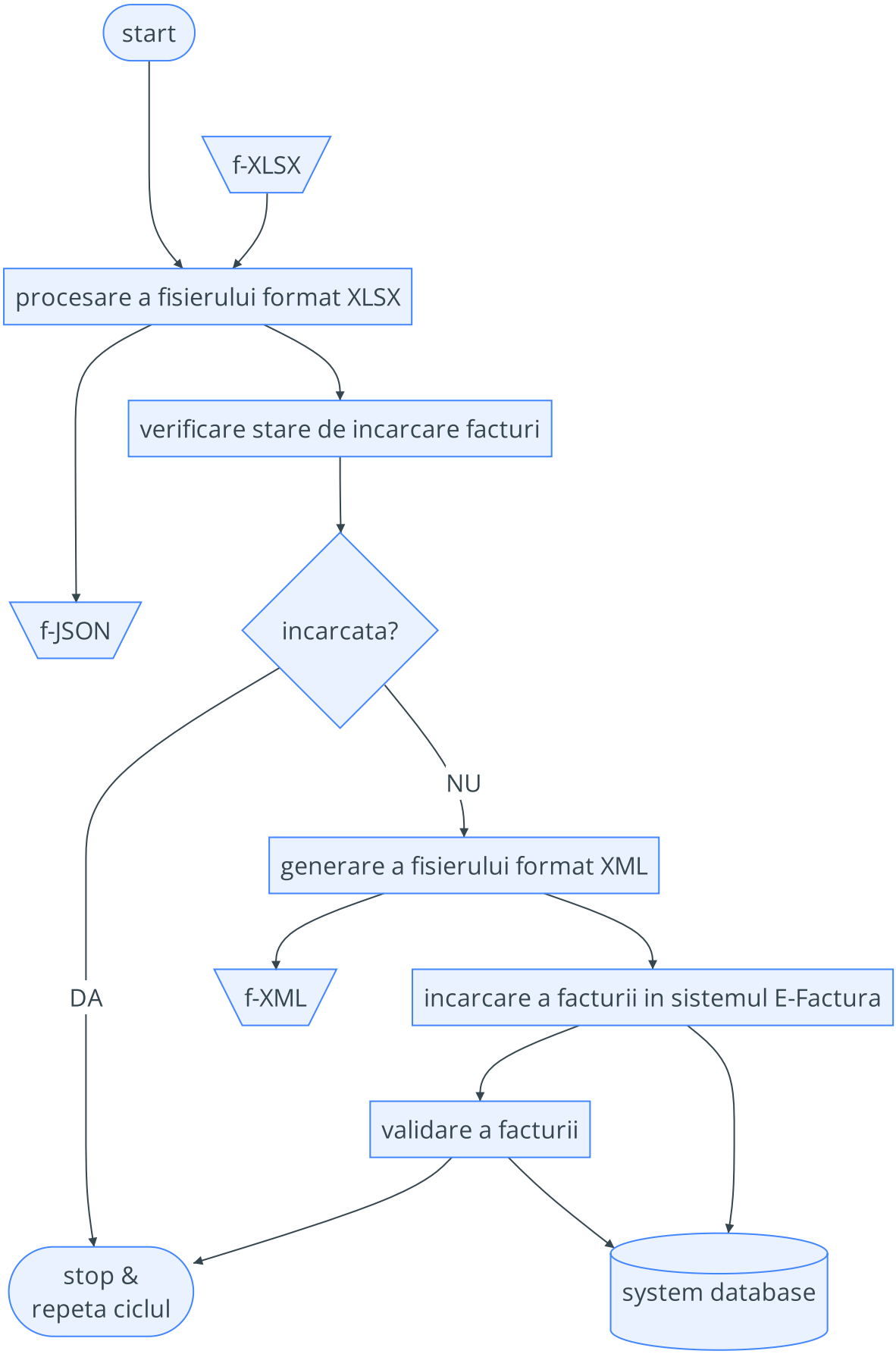
- **(RDINV)** modul de procesare a fisierului format `XLSX` ce contine factura si colectare a datelor aferente
  - *INTRARI*: fisier format `XLSX` ce contine factura emisa (cod: **`f-XLSX`**)
  - *IESIRI*: fisier format `JSON` imagine a datelor facturii (cod: **`f-JSON`**)
- **(WRXML)** modul de generare a fisierului format `XML`
  - *INTRARI*: fisier `f-JSON`
  - *IESIRI*: fisier format `XML` conform cerintelor si sistemului `ANAF E-Factura` (cod: **`f-XML`**)
- **(CHKXML)** modul de validare a facturii in sistemul `ANAF E-Factura`
  - *INTRARI*: fisier `f-XML`
  - *IESIRI*: raport cu eventualele erori de validare [1]
- **(LDXML)** modul de incarcare a facturii in sistemul `ANAF E-Factura`
  - *INTRARI*: fisier `f-XML`
  - *IESIRI*: raport cu validarea si identificatorul incarcarii [1]

- **(CHKISLD)** modul de verificare a starii de incarcare a unei facturi emise
  - *INTRARI*: fisier `f-XLSX` sau numarul / cheia / codul facturii
  - *IESIRI*: valoarea echivalent `TRUE` daca factura a fost deja incarcata sau valoare echivalent `FALSE` daca factura nu a fost incarcata [2]

Formatul fisierelor Excel cu factura

`XLSX` este sigurul format de fisier acceptat

27.3.1 Diagrama logica de functionare a componentei



...#FIXME explicatii necesare ?...

27.4 Componenta WEB_DASHB

...INCOMING...

27.5 Componenta SYSTEM_DB

...INCOMING...

---

1. raportul se scrie in baza de date a sistemului si in fisierul `f-XLSX` intr-un worksheet separat dedicat acestui scop ↵ ↵
2. in cazul valorii echivalent `TRUE` se poate intoarce identificatorul incarcarii daca este disponibil ↵

RENware Software Systems

28 Propunere tehnica sistem PayValidaBoa

Cuprins:

- p/n: `0000-0094`
- code-name: `payments_validation_board`
- commercial name: **PayValidaBoa**
- url propunere tehnica: `http://apitoroefact.renware.eu/commercial_agreement/110-SRE-payments_validation_board_req.uirements.html`
- git: `n/a`

Cuprins:

- Propunere tehnica sistem PayValidaBoa
  - Obiective
  - Vedere de ansamblu a solutiei
  - Cerinte functionale generale
  - Componenta xxx

28.1 Obiective

Acest sistem asigura prezentarea unui "*dashboard*" cu lista facturilor primite si starea lor de **verificare si aprobare interna** in vederea ordonantarii lor la plata.

28.2 Vedere de ansamblu a solutiei

Sistemul `payments_validation_board` consta din urmatoarele componente:

- **`INV_TOPMNG_BOARD`** - aceasta componenta prezinta *pentru MANAGEMNTul tip CFO* lista facturilor primite si starea lor referitor la *validarea si aprobarea lor finala* si un control pentru *APROBARE FINALA sau BUN DE PATA*.
- **`INV_CHK_BOARD`** - aceasta componenta prezinta *pentru VERIFICATORI si APROBABTORI* lista fa/turilor primite si *diverse controale pentru aprobarea si scrierea de diverse note* si observatii.
- **`INV_NOTIF_BOARD`** aceasta componenta prezinta notificari referitoare la *diversele OBSERVATII si NOTE* facture asupra facturilor primite in diverse stadii de aprobare de catre persoanele care efectueaza verificari asupra lor (prin componenta `INV_CHK`). Notificarile sunt disponibile atit in interfata aplicatiei iar *unele din ele pot fi transmise prin mail*.
- **`INV_LD_FOR_APPRV`** - aceasta componenta permite *incarcarea facturilor in fluxul de aprobare*. Optiuni de incarcare:
  - manuala (dintr-un board al aplicatiei)
  - automata dintr-un director
  - dintr-o baza de date externa sistemului (cu "marcarea" facturilor ce vor trebui incarcate)
  - la incarcare (indiferent de metoda) vor trebui specificati (sau dedusi din alte informatii) DESTINATARII ce primesc documentul
- **`ADMIN_CFG`** aceasta componenta permite *pentru ADMINISTRATORI* diverse *OPTIUNI DE CONFIGURARE*:
  - lista utilizatorilor ce fac parte dinfluxul de aprobare
  - adresele e-mail ale utilizatorilor
  - rolul utilizatorilor in accea ce priveste fluxul de aprobare facturile primite
  - certificate si semnaturile de certificare a "semnaturilor" de aprobare
  - ...

Figura urmatoare prezinta schematic rolul general al componentelor precum si interactiunea acestora cu mediul exterior sistemului `payments_validation_board`.

...IN PROGRESS...

28.3 Cerinte functionale generale

...INCOMING...

28.4 Componenta xxx

...INCOMING...

RENware Software Systems

29 Optiuni tehnice

Cuprins:

Acest document prezinta posibilele optiuni tehnice la cele doua sisteme, optiuni care vor trebui agreate si (preferabil) planificate cel putin din punct de vedere al prioritatii.

29.1 Optiuni sistem INVOICEtoROefact

Forma de utilizare si interactionare

- ☐ (**INVOICEtoROefact-RQ-01**) - varianta CLI (command line) cu utilizare "individuala"
- ☐ (**INVOICEtoROefact-RQ-02**) - varianta WEB cu utilizare centralizata

Configurabilitate

- ☐ (**INVOICEtoROefact-RQ-03**) - varianta in care se prelucreaza un model de fisier `Excel` in care sunt "fixate si blocate" locatiile celulelor ce contin date relevante
- ☐ (**INVOICEtoROefact-RQ-04**) - varianta in care structura si formatul fisierului `Excel` contin "cuvinte cheie" ce determina regasirea date relevante (de exemplu textul "Client:" intr-o celula semnifica inceperea unei zone cu datele clientului de la acea celula in jos si pina prima celula necompletata ce va fi gasita)

29.1.1 Recomandari sistem INVOICEtoROefact

- varianta (INVOICEtoROefact-RQ-01) este recomandata ca fiind "aproape obligatorie" deoarece chiar si in varianta WEB ea va trebui scrisa intr-o forma neutilizabila direct (sub forma de functie a sistemlui). Transformarea acestei functii in varianta CLI va permite o executie portabila ("la purtator") si offline (in situatii extreme se poate folosi doar fisierul `XML` generat si acesta va putea fi manual incarcat in ANAF-SPV). Informatia privind "starea de incarcare a facturii" va fi oricum salvata si in fisierul Excel aferent facturii si va putea fi preluata de catre varianta WEB pentru centralizarea informatiilor- a se vedea si modulul `LDXML` si notele de subsol aferente
- optiunea (INVOICEtoROefact-RQ-03) este recomandata ca varianta de start deoarece va permite realizarea unei variante OPERATIONALE (de lucru curent si testare) intr-un termen mai scurt, urmind ca aceasta optiune sa fie gradat extinsa si cu optiunea (INVOICEtoROefact-RQ-04). Aceasta "linie de lucru" nu va induce probleme, avind in vedere ca orice optiune / varianta aleasa ca varianta de start si planificata a fi extinsa va implica si MIGRAREA datelor deja produse la momentul extinderii ei

29.2 Optiuni sistem PayValidaBoa

Framework standardizat de orchestrare

- ☐ (**PayValidaBoa-RQ-01**) - utilizarea unui framework specializat de orchestrare si integrare cu alte sisteme "externe"
  - *Implicatii:* poate mari durata de implementare
  - *Avantaje:* utilizarea ulterioara pentru integrare intre sisteme ce prezita interfata standardizata (REST, SOA, NTFS, EXT4, OAuth, ...)

Semnare electronica a facturilor verificate

- ☐ (**PayValidaBoa-RQ-02**) - utilizarea de certificat tip "semnatura electronica" pentru autentificarea verificarii facturilor *EMBEDDED IN FACTURA*
- ☐ (**PayValidaBoa-RQ-03**) - utilizarea de certificat tip "semnatura electronica" pentru autentificarea verificarii facturilor *adiacet facturii - disponibil pentru consultare numai in sistemul PayValidaBoa*
- ☐ (**PayValidaBoa-RQ-04**) - verificarea facturilor nu necesita certificat tip "semnatura electronica" ci simpla informatie existenta in sistemul PayValidaBoa este suficienta

Baza de date 'interna / specifica' sistemului PayValidaBoa

- a se vedea documentul *"Considerente tehnice generale"*, sectiunea *"Considerente generale privind bazele de date proprii sistemelor"*, item *"(DBS-03)"* pentru opptiuni privind baza de date ce va fi utilizata "pentru operatiuni interne si specifice" de catre sistemul PayValidaBoa

29.3 Optiuni generale de implementare

Sistemele tip infrastructura ce vor fi utilizate

- ☐ (**general-RQ-01**) - server web-HTTP pentru aplicatiile de tip WEB (*ATENTIE:* sistemele WEB ce vor fi implementate necesita interfata / mod de operare standard `WSGI`)
- ☐ (**general-RQ-02**) - sistem de autentificare utilizat (intern aplicatie, Google, Identity Management propriu, ...)
- ☐ (**general-RQ-03**) - sistemele vor rula pe infrastructura proprie sau aceastea vor rula in infrastructuri gazduite

Alte optiuni 'ad-hoc' (in sedinta)

- ...
- ...
- ...
- ...
- ...
- ...

RENware Software Systems

30 Descarcare resurse

De retinut

Livrabile care nu sunt listate pentru o versiune sau care sunt declarate "InDISPONIBIL" nu sunt relevante si nu aduc nici o modificare fata de versiunea anterioara. Acestea exista de obicei doar in varianta *"developer" (ca surse)* si la nivel de cod sursa sunt relevante. Astfel acestea se recomanda a fi instalate doar pentru dezvolara specifice / integrari.

30.1 xl2roefact

Aici gasiti resurse aferente componentei: aplicatii, fisiere de configurare si alte elemente utile in utilizarea curenta.

30.2 web2roefact

Componenta indisponibila in aceasta versiune.

30.3 Sablon template factura

VI. About

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007
```

Copyright (C) 2007 Free Software Foundation, Inc. https://fsf.org/ (https://fsf.org/) Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

```
                            Preamble
```

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

```
                    TERMS AND CONDITIONS
```

1. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

1. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

1. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

1. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

1. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

```
a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7.  This requirement modifies the requirement in section 4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy.  This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged.  This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display
Appropriate Legal Notices; however, if the Program has interactive
interfaces that do not display Appropriate Legal Notices, your
work need not make them do so.
```

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

1. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

```
a) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by the
Corresponding Source fixed on a durable physical medium
customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by a
written offer, valid for at least three years and valid for as
long as you offer spare parts or customer support for that product
model, to give anyone who possesses the object code either (1) a
copy of the Corresponding Source for all the software in the
product that is covered by this License, on a durable physical
medium customarily used for software interchange, for a price no
more than your reasonable cost of physically performing this
conveying of source, or (2) access to copy the
Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the
written offer to provide the Corresponding Source.  This
alternative is allowed only occasionally and noncommercially, and
only if you received the object code with such an offer, in accord
with subsection 6b.

d) Convey the object code by offering access from a designated
place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no
further charge.  You need not require recipients to copy the
Corresponding Source along with the object code.  If the place to
copy the object code is a network server, the Corresponding Source
may be on a different server (operated by you or a third party)
that supports equivalent copying facilities, provided you maintain
clear directions next to the object code saying where to find the
Corresponding Source.  Regardless of what server hosts the
Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided
you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.
```

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

1. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

```
a) Disclaiming warranty or limiting liability differently from the
terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal
Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or
authors of the material; or

e) Declining to grant rights under trademark law for use of some
trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that
material by anyone who conveys the material (or modified versions of
it) with contractual assumptions of liability to the recipient, for
any liability that these contractual assumptions directly impose on
those licensors and authors.
```

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

1. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

1. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

1. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

1. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

1. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

1. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

1. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

1. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

1. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

```
                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs
```

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and` show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see https://www.gnu.org/licenses/ (https://www.gnu.org/licenses/) .

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read https://www.gnu.org/licenses/why-not-lgpl.html (https://www.gnu.org/licenses/why-not-lgpl.html) .

VI.I Despre INVOICEtoROefact

RENware Software Systems

INVOICEtoROeFact Project

31 CHANGELOG

- For version code structure meaning see SDEVEN methodology document (http://sdeven.renware.eu)
- `<PROJECT ROOT>/doc_src/` is the default starting location in a file path (if not clear from context) (**ATTN** - in production environment is `docs/`)
- `<WEB_ROOT>/` is the HTTP server root directory, as default `docs/` and supposed if no other parent is specified

31.1 `0.9` init commands layer 2 of functionalities

- tbd... move `xl2json` functionality to `l2_commands.py`
- tbd... move `settings` functionality to `l2_commands.py`

- `upd-Rdoc` update `README_xl2roefact_library.md` make a small hierarchical diagram with component layers
- `upd-DLD` upd all `xl2json` docstring and generate new DLD doc
- `main-code-cls` close `xl2json` method by prep status result before owner file not valid exiting & close all open TODO_FIXME issues
- `main-code-ini` create `commands.py` to accommodate *layer 2 commands functionalities*
- update version to `0.9.dev0`

31.2 `0.8` BUGFIX xl2roefact entry point

- upd xl2roefact dowloads & build site
- make WHEEL deliverables: build, PyPi publish
- created `gitupd_tags.sh` to update all local tags keeping only remote ones & made it execurable
- updated versions
- BUGFIX updated `xl2roefact.setup.py` entry point for `xl2roefact` to correct to `xl2roefact.app_cli...` from `src.app_cli...`

31.3 `0.7` clean xl2roefact package and invoice JSON

- `update downloads` ref 0.7 dlvbs & reorganize it with 1st level. Follow:
  - ref new structure & locations of download docs
  - ✅ update master downloads.md ref old versions drop
  - ✅ links: add `0.7`
  - ✅ remove all links `<= *0.5*`
  - ✅ removed all deliverables `<= *0.5*`
- build & publish deliverables + site
- build & publish site with new downloads structure
- `brk-doc_src/downloads.md` letting it as *master* and making specialized download pages in `xl2roedact/doc/package_downloads.md` and `excel_invoice_template/package_downloads.md`:
  - ✅ updated new downloads files for existing deliverabkes
  - ✅ created directories structure & empty files
- update version & site
- `xl2roefact-refactor`
  - fixed `xl2roefact` command `xl2json` option `-o`, set ver to `.dev`, rebuilt dld doc and site
  - reverted to flat package structure, without src/ directory but only xl2roefact/

  - updated PDM scripts and made package from src/ directory (created **init**.py)
  - refactored xl2roefact source files
- `upd-site` updated sitem left side navigator to better reflect the new features / system architecture
- `fix-site` 0.7rc2 downloads.md, link to "Descarcare sablon de fisier de configurare"

31.3.1 `0.7rc2` updated console application to run in silent or vebosed

- updated downloads.md & built site
- review & update DLD doc, build all deliverables and publish on PyPi
- `app_cli.xl2json` updated to write function out at verbose or otherwise (ie, not verbose) just its print messages. Also eliminate the JSON printing when verbose because no more debug necessary at this moment
- `rdinv-silent` updated `rdinv.rdinv()` in order to run "in silent" and to emit all print info in a specified parameter not None, or (if parameter is None or not specified) emmit normally to `stdout` device
- `init-command-layer` made `commands/` as in-package / layer
- upd versions before start work

31.3.2 `0.7rc1` review & clean `xl2roefact.rdinv` module of TODOs

- a. publish PyPi, upd downloads.md, site
- a. build only wheel deliverable because only source library is impacted
- a. review TODOs, keep only those that are future features and move them to ROADMAP
- a. upd versions before start work

31.3.3 `0.7rc0` settings refresh option and app settings reload by request

- update downloads, site
- Published on PyPi (`https://pypi.org/project/xl2roefact/0.7rc0/`)
- updated doc, deliverables
- `config-settings-main-func` update config_settings module to embed init code under a `main()` function which run @ init but can be latter called also
- `240429piu01` update doc `README_xl2roefact_library.md` section "Library components" to reflect the new commands layer. Rebuild site & publish
- fix `invoicetoroefact.renware.eu` site for 0.6 MSI & EXE downloads

31.4 `0.6` invoice type JSON-XML tag

- `upd-dwnlds` update downloads.md. site rebuild & publish
- `pypi-publish` publish on PyPi
- `xl2roefact-build` build all `0.6` deliverables
- `app-readme-doc` check & update xl2roefact `README.md`:
  - ✅ example JSON schema update & build + publish site
  - ✅ `doc-sys-settings-feat` short note about allowed invoice type (`cbc_InvoiceTypeCode`)
  - ✅ `doc-sys-settings-feat` in xl2roefact library doc, ie "Referinta dezvoltare sofrware" (`README_xl2roefact_library.md`) explain how manage system settings using `sys_config.py`
  - ✅ `for-dvelopers-section` introduce a new section "Referinta dezvoltare sofrware" to group existing and all new things ref xl2roefact library. Reference to existing `README_xl2roefact_library.md`
  - ✅ `site-0.6.dev1` restructured whole design presentation for a better view of its architecture: updated from a pure technical view to end-user technical view
  - ✅ `240413_01` updated "API Reference" bullet link (from begging of doc) to point directly to published site as intended for `PyPi` availability in project description
- xl2roefact change version
- `refact-xl2roefact-modules-dirname` refactoring xl2roefact modules directory name to `src/` (old was `xl2roefact/`)
- `InvoiceTypeCode-app-param` make `invoice_type_code` choose-type app parameter
- `InvoiceTypeCode-func-param` make `xl2roefact.rdinv()` parameter `invoice_type_code` parameter with default value `InvoiceTypesEnum.NORMALA`

31.4.1 `0.6rc0` system database and parameters

- `0.6rc0+240420` build and publish release:
  a. update versions for `xl2roefact` and `invoicetoroefact`
  b. build xl2roefact documentation
  c. wheel deliverables build & PyPi publish
  d. updated `downloads.md`
  e. site build & publish
- `sys_settings-invoice-type` populated "system database" with allowed invoice types
  - created `InvoiceTypes` dictionary with allowed invoice types
  - created `InvoiceTypesEnum` as Enum to be used by CLI app parameter (dynamically generated from previous data object)
- `sys_settings-module` created `xl2roefact/sys_settings.py` component dedicated to system settings (ie, not user configurable but only developers; is intended that later versions to use also a database for)

- `cbc_TaxPointDate` will be set to 25 of next month from invoice issued month
- `cbc_DueDate` search `invoice_header_area` ref `PATTERN_FOR_DUE_DATE` pattern. Use found data if not None or default it to `invoice_header_area["issued_date"]["value"] + DUE_DATE_DAYS` if None found
- `PATTERN_FOR_DUE_DATE` update `config_settings.py` & `app_settings.yml`, create `PATTERN_FOR_DUE_DATE = ["scad", "due da", "date due"]`
- `inv-issdate-todate` upd `rdinv` for final json dict convert & local save invoice issued date in `datetime` format to--> `tmp_reusable_items["invoice_issdate_asdate"]`
- `cac_Delivery` set as invoice issued date
- `cac_PaymentMeans` will be set to `1` supposing is unknown at invoicing issuing date
- `DEFAULT_DUE_DATE_DAYS` new app config parameter with default value 30 days
- `cbc_Note` set to "processed @ `{date_time_now}` with xl2roefact". Latter this field will be updated with text ref loading to RO-eFact data-time
- `init-work` set site & xl2roefact versions to `0.6rc1`

31.4.2 `0.6.dev1` code missing XML tags

- `arch-prev-rlse-chlogs` archive `0.5.4` CHANGELOG
- `xml-json-map` updated `xl2roefact.rdinv` module for XML-JSON map
- `fin-xml-specs` made `xl2roefact/tests/todosXML.md` file with list of XML tags to do and all other specs to complete activity
- `init-work` set site & xl2roefact versions to `0.6.dev1`

31.4.3 `0.6.dev0` clean xl2roefact & invoicetoroefact projects (...yymmdd hhmm...)

- `240408piu-adm1` cleaning and updating version strings and code
  - rebuild site
  - update xl2roefact/**version.py**
  - update main versions.yml
- `240408piu-adm1` cleaning and updating environments:
  - updated xl2roefact python requirement, relaxed to `>=3.10`
  - updated site version to `0.6.1dev0` to mark in progress work
  - installed `chromium` on dev server

32 Archived CHANGELOGs

32.1 0.5 version

- `0.5.4` invoice supplier from owner master data
- `0.5.3rc1` fix invoice JSON key "cac:Party" naming
- `0.5.3rc0` invoice supplier from Excel
- `0.5.2.dev2` release xl2roefact. `0.4.1.dev1` fix sEXE bug from `0.4.1.dev0` version
- `0.5.1.dev1` site readability improvements

32.2 0.4 version

- `0.4.1.dev0` xl2roefact include a data directory in package for various data files "built-in" package
- `0.4.0.dev2` externalize recommended rules for updating app setting rules

32.3 0.3 version

- `0.3.2b0` single EXE version
- `0.3.1b1` fixed bug JSON->["Invoice"]["cac_InvoiceLine"] list[list]
- `0.3.1b` promote v0.3.0b0 deliverables: WHEEL, TRA.GZ, MSI to `0.3.1b`
- `0.3.0b` xl2roefact invoice taxes summary

32.4 0.2 version

- `0.2.2.dev` project development environment improvements
- `0.2.1b` invoice grand totals
- `0.2.0b` xl2roefact invoice customer info-optional items (bank, email, reg-com, phone)

32.5 0.1 version

- `0.1.22b` xl2roefact application interface improvements
- `0.1.21.post3` cleaned system documentation and site
- `0.1.21.post2` xl2roefact app detailed section with commands & options "--help" like
- `0.1.21.post1` fixed missing links in site root index page
- `0.1.21` rollout news in system portal invoicetoroefact.renware.eu
- `0.1.20.dev` invoice customer address
- `0.1.19.dev` invoice customer and partial invoice total values calculations
- `0.1.18.dev` invoice customer CUI partial invoice total values calculations
- `0.1.17.dev` fixed all application & package running standard ways
- `0.1.16.dev` improving Excel kv-data search with "IN-LABEL" method
- `0.1.15` updated solution portal `http://invoicetoroefact.renware.eu/`
- `0.1.14.dev` invoice issue date
- `0.1.13.dev` invoice currency
- `0.1.12.dev` invoice number
- `0.1.11.dev` packaging improvements for app & xl2roefact package
- `0.1.10.dev` command interface improved, `msi` package building, invoice template & updated documentation
- `0.1.9.dev` `xl2roefact.RDINV` running executable and distribution kit
- `0.1.8.dev` improved application structure and first executable release
- `0.1.7.dev` `xl2roefact.RDINV` invoice items & metadata + *OPEN ISSUES*
- `0.1.6.dev` commercial agreement OPTIONS document
- `0.1.5.dev` init component *xl2roefact* for CLI application
- `0.1.4.dev` Create system backbone structure
- `0.1.3.dev` Enhancing `payments_validation_board` technical proposal
- `0.1.2.dev` Enhancing `APItoROefact` technical proposal
- `0.1.1.dev` Elaborating technical proposal
- `0.1.0.dev` System raw backbone