

## RENware Software Systems

- [xl2roefact](#)
  - [Facilitati \(Features\)](#)
  - [Instalarea](#)
    - [Instalarea aplicatiei xl2roefact](#)
    - [Instalarea bibliotecii Python \(package\) xl2roefact](#)
  - [Configurarea aplicatiei xl2roefact](#)
  - [Comenzile aplicatiei](#)
    - [about](#)
    - [settings](#)
    - [xl2json](#)
  - [Practici si reguli referitoare la continutul facturilor din Excel](#)
  - [Tutorial utilizare aplicatie](#)
    - [Organizarea informatiei](#)
    - [Exemplu de procesare a unei facturi](#)
  - [Formatul fisierului JSON](#)
  - [Documentatia tehnica](#)
  - [Descarcare \(download\) aplicatie xl2roefact CLI](#)
    - [Arhiva versiuni publicate disponibile](#)

## xl2roefact

versiune MSI installer	0.1.19
------------------------	--------

versiune script Python	0.1.20
------------------------	--------

versiune pachet Python PyPi	0.1.20
-----------------------------	--------

## Facilitati (Features)

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

(This application is all about electronic invoices creation from Excel office format (xlsx). It can generate invoice in JSON, XML, PDF formats and even upload it to *RO E-Fact* system.)

- Components
  - `x12roefact` **command line application** to create, manipulate and upload to RO E-Fact system Excel invoices
  - `x12roefact` **Python library** to use functions to create, manipulate and upload to RO E-Fact system Excel invoices
- Identification
  - code-name: `x12roefact`
  - copyright: (c) 2023, 2024 RENWare Software Systems
  - author: Petre Iordanescu (petre.iordanescu@gmail.com)
- Deployments
  - Windows: `x12roefact.exe` 64 bit CLI application (installable through a `MSI` package)
  - Linux: `x12roefact` executable CLI shell
- Specifications
  - command general format: `x12roefact [COMMAND] [OPTIONS]`
  - help: `x12roefact [COMMAND] --help`

## Instalarea

Acest sistem consta din urmatoarele componente:

- o aplicatie de tip *linie de comanda / consola* ce permite realizarea operatiilor necesare de procesare a facturilor in Excel ( `x12roefact` CLI application)
- o biblioteca ( `x12roefact` ) tip "*standard Python package wheel*" ce permite utilizarea functionalitatilor de procesare a facturilor Excel in mod programatic in alte sisteme. Biblioteca este realizata in Python 3 iar instalarea ei se poate face cu instrumentele standard Python

### Instalarea aplicatiei x12roefact

Pachetele de instalare se gasesc in directorul `dist/` ca arhive `ZIP`. Pachetele disponibile contin in numele lor versiunea de aplicatie utilizata si versiunea sistemului de operare pentru care sunt disponibile. (EN: *Installation*

package is found in `dist/` directory in archive (zip) files. Available packages are identified by released versions and operating systems.)

Pachetele contin un script de instalare sub forma standard `MSI` pentru Windows si `DEB` pentru Linux Debian (verificati disponibilitatea pentru sistemul de operare folosit de dvs). (EN: Packages contains an installation script as `MSI` for Windows and `DEB` for Debian Linux based systems (check availability for your operating system)).

## Instalarea bibliotecii Python (package) xl2roefact

Instalarea acestuia se face cu instrumentele standard Python. Recomandarea este pentru instalarea simpla cu: `pip install xl2roefact`, biblioteca fiind disponibila in repository-ul standard PyPy. Pentru instalarea din surse, biblioteca poate fi descarcata din [GitHub](#).

## Configurarea aplicatiei xl2roefact

Parametrii de configurare a aplicatiei se gasesc in fisierul `config_settings.py`. Acestia sunt sub elaborati in limbaj Python prin utilizarea conventiilor de constante conform recomandarilor PEP (numele capitalizat) si sunt acompaniati 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
```

## 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
```

### about

Afiseaza informatii despre aceasta aplicatie (copyright, scop, etc).

### settings

Afiseaza parametrii de configurare a aplicatiei. [Vezi sectiunea de configurare a aplicatiei](#).

## xl2json

Transforma fisierul (fișierele) Excel în forma JSON pentru utilizare ulterioară ca formă de date standardizată pentru schimbul de informații cu alte sisteme electronice. Formatul JSON utilizat conține:

- informațiile aferente facturii (cheie: `Invoice`)
- o hartă de ajutor în conversia formatului JSON în formatul XML acceptat de sistemul RO E-Fact (cheie `meta_info.map_JSONkeys_XMLtags`) și definițiile XML aferente (cheie `meta_info.invoice_XML_schemes`)
- alte informații despre fisierul Excel prelucrat (alte chei din `meta_info`)
- datele preluate din formatul original Excel (cheie `excel_original_data`) - acestea sunt utile pentru depanare în caz că aceasta este necesară în cazul specific al fișierului Excel folosit de dvs

## Practici și reguli referitoare la conținutul facturilor din Excel

Acest capitol se referă la modul în care este "tratat" conținutul fișierului Excel cu factura, mai exact la modalitatea în care informația facturii este căutată, identificată și găsită în scopul de a fi salvată în oricare din formatele de "factura electronică / E-Fact".

Utilizarea șablonului de factură Excel ce este livrat împreună cu aplicația **ESTE O VARIANTA DE LUCRU RECOMANDATA**, dar nu obligatorie. Chiar și în cazul utilizării acestuia, prin modificarea "structurii" acestuia, informația poate ajunge *nerecognoscibilă / neidentificabilă* total sau parțial dacă nu sunt urmate regulile expuse.

În general trebuie făcută diferență între datele facturii și modul în care aceasta va fi tipărită (va apărea la tipărire / previzualizare).

Mai exact **conținutul informational** al facturii nu trebuie nici confundat și nici mixat cu **formatul de afișare al acesteia** (layout). Pentru acesta din urmă se recomandă a fi folosite cu precădere *regulile de formatare* din Excel și nu cele stocare a datelor. Un exemplu este un număr zecimal oarecare unde:

- una este valoarea introdusă într-o celulă (de ex cu 3 zecimale) și
- alta este valoarea afișată (cu 2 zecimale) - aceasta din urmă trebuie obținută prin formatarea celulei respective de a afișa 2 zecimale prin rotunjire întrucât valoarea efectivă trebuie să fie cea originală cu 3 zecimale, lucru (diferența) care se poate vedea la editarea conținutului celulei.

-#TODO ...tbd (se va lua din `_docstring_` aferent `config.settings.py`)

## Tutorial utilizare aplicație

### Organizarea informației

Aplicația *xl2roefact* "promovează" structurarea informației procesate astfel încât să fie evitată situația "*de aglomerare*" a directorului curent cu fișiere ce trebuie identificate și izolate în situația în care se fac *procesări în masă* (pe mai multe fișiere / facturi sursă).

Astfel, aplicatia se asteapta 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:

```
x12roefact -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: `x12roefact -d ./`
- procesarea tuturor facturilor facturilor din luna *iunie*, copiate intr-un director dedicat sub directorul curent:  
`x12roefact -d ./facturi_iunie/`

#### Exemplu de procesare a unei facturi

- se creaza directorul `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: `x12roefact x12json`

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

## Formatul fisierului JSON

Structura de baza a fisierului JSON aferent unei facturi este:

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

Cheile de la primul nivel reprezinta:

- **Invoice** - #TODO tbd...
- **meta\_info** - #TODO tbd...
- **excel\_original\_data** - #TODO tbd...

## Documentatia tehnica

### Descarcare (download) aplicatie xl2roefact CLI

- 0.1.20.dev invoice customer address [MSI installer win64](#)

### Arhiva versiuni publicate disponibile

- 0.1.19.dev invoice customer and partial invoice total values calculations [MSI installer win64](#)
- 0.1.18.dev invoice customer CUI partial invoice total values calculations [MSI installer win64](#)

**NOTA:** Pentru descarcarea bibliotecii, pachetului Python `xl2roefact` [procesul este descris aici](#)

---

Last update: January 29, 2024