EDI Manager

The EDI Manager is a Rust-based system for integrating MoneyWorks (MW) accounting software with the Peppol/Edimondo e-invoicing and e-ordering network. It automates the import, export, and processing of invoices and orders between Moneyworks and Edimondo using UBL (Universal Business Language) XML documents.

Technical documentation

Key features:

- **MW Transaction Export:** Extracts transactions from Moneyworks, builds UBL XML documents using JSON templates, and exports them to Edimondo.
- **MW Transaction Reprocessing:** Supports reprocessing of transactions based on status and colour flags.
- **Edimondo Status Checking:** Checks the delivery status of sent documents and updates MW/DB accordingly.
- **Order Import:** Retrieves orders from Edimondo, parses UBL XML, and imports them into MW.
- **Configurable:** Uses TOML configuration files for server, business, and email settings.
- **Logging:** Logs operations and errors to rotating log files.
- **Utilities:** Provides date/time formatting, file writing, and folder setup functions.

Main modules:

- mw_to_ubl.rs: MW → UBL/Edimondo export logic.
- ubl_to_mw.rs: Edimondo → MW import logic.
- mw.rs: MW API communication.
- edimondo.rs: Edimondo API communication.
- invoice.rs, order.rs: UBL document construction and parsing.
- export.rs, operations.rs: MW transaction and detail handling.
- db.rs: Database operations for exports/imports.
- config.rs: Configuration management.
- utilities.rs: Utility functions.

Entry point:

main.rs calls `start_server` which runs the main service loop.

Data flow:

Moneyworks transactions are exported, transformed to UBL XML, sent to Edimondo, and their status is tracked. Incoming orders from Edimondo are parsed and imported into Moneyworks.

Config files:

- config.toml for settings
- control_file.txt for last modified timestamps
- JSON templates for UBL documents in templates

For more details, see the main service logic in lib.rs and mw_to_ubl.rs

Configuration Settings

The configuration file (config.toml) controls all runtime behaviour. The main sections and their fields are:

```
====  MoneyworksServerConfig ====
* `url`: Base URL for Moneyworks REST API.
* `folder_name`, `folder_pw`: Moneyworks data folder and password.
* `document_username`, `document_pw`: Moneyworks document user credentials.
* `file_name`: MW database file name.
```

```
====  BusinessInfo ====
* `party_endpoint_id`: Peppol endpoint ID for your business.
* `party_identification`: Business registration number.
* `party_name`: Legal name of your business.
* `street_name`, `additional_street_name`: Address details.
* `city_name`, `postal_zone`, `country`: Location details.
* `tax_scheme`: Tax scheme identifier.
```

```
====  EdimondoConfig ====
* `url`: Edimondo API base URL.
* `api_key`: API key for Edimondo authentication.
* `outbox_faktura_boxid`: Box ID for outgoing invoices.
* `inbox_ordrer_boxid`: Box ID for incoming orders.
```

```
====  OutputFolders ====
* `invoices_out_folder`: Path for saving generated invoice XML files.
* `orders_in_folder`: Path for saving imported order XML files.
```

```
====  Settings ====
* `reexport_colour`: Moneyworks transaction colour used to mark transactions
```

```
for reprocessing.
* `database_url`: SQLite database connection string.
```

```
====  Email ====
* `host_name`: SMTP server hostname.
* `from_name_address`: Sender email address.
* `email_address`: Login email address.
* `password`: SMTP password.
* `port_number`: SMTP port.
* `send_to_addresses`: List of recipient email addresses for notifications.
```

Note that the config file contains passwords and keys, so you should not send it via email or store it at any other location unless the data is encrypted and password protected.

From:
https://www.docs.sussol.net/ - **Sussol Docs**

Permanent link:
https://www.docs.sussol.net/doku.php/moneyworks:edi_manager?rev=1759894356

Last update: **2025/10/08 03:32**