## Project Overview

The DMH\_MR\_Tool is a lightweight automation utility designed to streamline the manual process of collecting, extracting, and processing Australian financial market data—such as distribution and component details—from external sources like ASX, iShares, BetaShares, Vanguard, or internal repositories.

It helps users identify relevant updates, extract key data from Excel or PDF reports, and input information into the internal DMH system based on client-specific business rules.

The tool aims to reduce daily processing time (from 2–3 hours to under 30 minutes) and minimize human error, improving both efficiency and data accuracy.

## Technical Architecture

* **Type**: Desktop application built with Python and PySide6.
* **Target Users**: Single-user environment; not designed for concurrent access.
* **Tech Stack**:
  + **UI**: PySide6
  + **Data storage**: SQLite3
  + **Async I/O**: aiohttp, aiosqlite
* **Deployment Strategy**:
  + **Code** is packaged via internal company macro into a .deploy file.
  + .deploy is uploaded to the **GMAS** platform.
  + End users download a .gmas file (~8KB) and run it via Automation Launcher.
  + **Automation Launcher** loads code from the cloud and runs it using a built-in Python interpreter.
* **Performance Considerations**:
  + Small-scale data volume (~10–20 records/day).
  + Local SQLite3 used for temporary storage.
  + Tool runs asynchronously where possible, but overall usage is light and single-threaded.
* **Compatibility Risks**:
  + Automation Launcher uses a predefined Python environment.
  + Dependency versions may differ from local dev setup, occasionally leading to runtime issues after deployment.

## Data Specifications

**Data Sources & Formats**

* **ASX**
  + **Announcement Metadata**:
    - Fetched via HTML scraping from:

[Announcement Search by ASX Code] https://www.betashares.com.au/fund/high-interest-cash-etf/#announcements

[Daily Announcements] <https://www.asx.com.au/markets/trade-our-cash-market/todays-announcements>

* + - Data includes announcement title, publish date, URL, and associated ASX code.
    - Parsed using XPath; stored in SQLite.
  + **Announcement Documents**:
    - Downloaded PDFs (e.g. distribution statements) based on metadata URLs.
* **Vanguard**
  + JSON API returns latest component data for all tickers.
  + Includes: ticker, component values, publish date, ex-date, etc.
  + Internal mapping used to align Vanguard-specific component names with DMH nomenclature.
* **BetaShares**
  + HTML scraped from fund-specific URLs (e.g., [https://www.betashares.com.au/fund/<fund-name>#announcements](https://www.betashares.com.au/fund/%3cfund-name%3e#announcements)).
  + Fund list (~20 funds) cycled through; each returns a list of announcements.
  + Parsed using XPath; similar structure to ASX.
* **iShares**
  + [To be integrated] — source and structure TBD.
* **Processing Logic & Update Frequency**
  + **ASX Daily Feed**: Queried once per day to ingest all new announcements.
  + **ASX by Ticker**: Queried on demand (typically ~10 lookups/day) based on DMH exceptions.
  + **BetaShares/Vanguard**: Queried periodically (~10 times/month) during expected reporting windows.
  + **PDF Downloads**:
    - Triggered when announcements are matched to DMH exception tickers.
    - Stored in a shared network drive and marked as "downloaded" in the announcement database.
    - PDF formats vary significantly across issuers; ASX only hosts documents and does not standardize their structure. The tool supports 5–6 common layout patterns and handles ~10+ edge cases using custom parsers.
* **Data Volume & Performance**
  + **Volume Estimates**:
    - ~100–200 announcement entries/day across all sources.
    - ~1–5 PDFs downloaded on regular days, with peaks of 10–30/day during monthly reporting periods.
  + **Growth Expectations**:
    - Data volume remains low and predictable.
    - No significant scaling concerns due to single-user desktop nature.
* **Storage & Retention Policies**
  + **Metadata (announcements)**:
    - Stored in a SQLite database hosted on a shared network drive.
    - Lightweight structure (ASX code, title, date, URL, download status).
    - Intended for long-term retention with no purge policy.
  + **Documents (PDFs)**:
    - Stored in a shared network drive.
    - Manually cleaned by developers once per year to conserve space.

## Database Schema Details

This tool uses a lightweight SQLite3 database to cache and track downloaded financial announcement metadata and related information. Below are the core tables and their structures.

* **Table** asx\_info

Stores metadata for announcements retrieved from the ASX website based on ASX code or daily queries.

CREATE TABLE asx\_info (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    asx\_code TEXT NOT NULL,

    headline TEXT NOT NULL,

    date TEXT NOT NULL,              -- Format: YYYY-MM-DD

    url TEXT NOT NULL,

    downloaded INTEGER DEFAULT 0     -- 0 = not downloaded, 1 = downloaded

);

## UI System

**Home Interface**:

Software Usage Dashboard/Overview

**File Interface:**

Show the latest data update time of each website in database (Card Area)

Download a single day's data from a custom website (Card Area – button)

Download data from a custom website using a custom ASX code (Card Area – button)

Activate daily spider process (Top area - button - Update the latest information from each website and download data required Send the data to backend parser and flow to the MR Update Interface if any data created by parser)

Show the database information, including table, DB address (Card Area)

**Parse Interface:**

Drop and Drop a file in a gray area, and select a template to start a File Parser process and display the result. The UI will display the template in a form first and the display the parsed content in the form.

Column Name 1 in DB | re pattern 1 | (value after parsed, can be edited) | comment

Column Name 2 in DB | re pattern 2 | (value after parsed, can be edited) | comment

Some columns are not directly parsed from the file, a certain number of business rules will apply on them, like it can be the sum of other columns. In this case, the applied business rules will be displayed in the Comment field.

Add an extra row from a dropdown list of column name, and input a value and comment

Submit the entire form (Use added row included) to MR Update Interface. User will be asked to input some extra information used to display in the MR Update Interface in a pop-out window (Bottom area - button)

Or ignore the drop and drop area, input a folder path in a text-input field. The files in the folder will automatically parsed and flow to the MR Update Interface.

**MR Update Interface:**

An editable table area will show, table will have a fixed column name set

Task data can be pasted in this area, including task info from Parse Interface

Submit this task form to DMH system (bottom Area – Button)

Real-time submitting status will display in the table

Click a single row in the table will show the specific business data of it

Automatically save the backup file when updating successfully (each task row will generate a backup file)

**DB Browser Interface:**

Query a remote Sqlite3 database by SQL (Top Area)

Query result will be displayed (Bottom Area)

Data in the query result table can be re-ordered and exported

**Manual Interface:**

Display the business logic of the MR Update Interface

**Setting Interface:**

Access the software log file in real time

Modify the log file’s path. download file path, backup-file saved path

## Log System:

Log error while executing python functions, display filename, function name, vars, code line when error raised, can be added in front of a function as a decorator (like loguru)

Log files will be saved in a certain path, and renew a file when reached 5mb

## Database System:

Table – asx\_headline

Table – nz\_asx

Table - mr\_update

Table - vanguard