Back Office Tools

User Guide

Version:1.1

Approval date:

Table of Contents

[Revision history 2](#_Toc49956131)

[1 Introduction 3](#_Toc49956132)

[2 System architecture 4](#_Toc49956133)

[3 User 4](#_Toc49956134)

[4 Features with GUI 4](#_Toc49956135)

[4.1 Bulk Novation 4](#_Toc49956136)

[4.2 ISIN Suppression for Finex asset prices 8](#_Toc49956137)

[4.3 ISIN Suppression for Finex target market data 9](#_Toc49956138)

[4.4 Update Marketing Campaign for Adviser 11](#_Toc49956139)

[5 Watchers 15](#_Toc49956140)

[5.1 Altus Transfer Gateway 15](#_Toc49956141)

[5.2 Asset Classification 17](#_Toc49956142)

[5.3 Asset Loader 18](#_Toc49956143)

[5.4 Bloomberg 18](#_Toc49956144)

[5.5 Calastone 19](#_Toc49956145)

[5.6 Distribution FINEX 20](#_Toc49956146)

[5.7 EMX Fund List 21](#_Toc49956147)

[5.8 Finex asset prices 22](#_Toc49956148)

[5.9 Finex target market data 24](#_Toc49956149)

[5.10 GAD and SMPI/ARS 25](#_Toc49956150)

[5.11 Generic Data Extract 27](#_Toc49956151)

[5.12 HSBC ACK Processing 32](#_Toc49956152)

[5.13 Reporting – ISACOM100 35](#_Toc49956153)

[5.14 Reporting – ‘Other interest Return (Type 18)’ 37](#_Toc49956154)

[5.15 Reporting – ‘Product Sales Data’ report 38](#_Toc49956155)

[5.16 Reporting – RPSCOM100: Sonata TXT to XLSX and XLSX to Sonata TXT 39](#_Toc49956156)

[5.17 Model Portfolio account linking processor 42](#_Toc49956157)

# Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Reason for Change** | **Approved by** |
| 25/10/2018 | V1 | Dominik Niszewski | First draft |  |
| 02/09/2020 | 1.1 | Dominik Niszewski | Added Asset Loader sample input file  Added mapping and transformation file and process steps file for Finex target market data processor |  |
|  |  |  |  |  |

# Introduction

The main purpose of the *Back Office Tools* (BOT) application is to provide functionality that is not available in the other systems and to enable the other systems to exchange information.

BOT comprises 4 web pages with GUI and 23 data processors.

The pages allow the user to:

1. maintain the list of assets to be excluded from Finex files (asset prices and target market) imported to Sonata,
2. reassign multiple clients and accounts from one adviser to another (bulk novation) at one go,
3. reassign multiple clients and accounts from one marketing campaign to another at one go.

The data processors participate in delivering:

1. information about Altus Transfer Gateway account transfers to Blue Button and Sonata systems,
2. asset data, including asset classification, to Sonata system,
3. Bloomberg asset prices to Sonata system,
4. Sonata orders to Calastone system,
5. order confirmations from Calastone to Sonata system,
6. information about dividends related to assets from Finex to Sonata system,
7. information about fund asset statuses from EMX to Sonata system,
8. Finex asset prices to Sonata system,
9. Finex target market attributes to Sonata system,
10. Dunstan Thomas illustration of pension prediction to Sonata and Print Service systems,
11. multiple files from Blue Button and Sonata to HSBC, Prudential and Scottish Friendly systems,
12. multiple files from HSBC, Prudential and Scottish Friendly to Blue Button and Sonata systems,
13. Sonata payment files to HSBC system,
14. information from HSBC acknowledgement files to the users,

Besides, they

1. amend Blue Button and Sonata versions of ISACOM100 report so that they can be manually merged,
2. amend Blue Button and Sonata versions of ‘Other interest Return (Type 18)’ report so that they can be manually merged,
3. merge Blue Button and Sonata versions of ‘Product Sales Data’ report.

Data processors transform files appearing in input folders, clear the input folders and place the resulting files in output folders. Processing is triggered by the appearance of the appropriate files in the input folders. In most cases the file transformation consists in some of the following:

1. changing file names and extensions so that the files can be loaded successfully,
2. splitting or merging the files so that each system receives the appropriate rows,
3. filtering out certain file rows so that irrelevant rows are ignored,
4. amending the values and sometimes the column structure in the files so that the content of the files can be loaded successfully.

Actions described in points b) – d) are based on certain formulae / filtering rules using values from the input file, values from external data sources and hardcoded values.

In addition to that some BOT processors:

1. trigger generation of notification emails,
2. start letter generation in Sonata.

As far as archiving and logging errors is concerned, data processors work in an analogous way:

1. input and output files are archived in dedicated Archive folders,
2. files in Archive folders contain timestamps and source folder names (the source folder for an input file is the folder BOT takes the file from for further processing whereas the source folder for an output file is the folder BOT puts the file in after producing it) in their names,
3. in case error occurs, files in error are copied to dedicated Errors folders,
4. files in Errors folders contain timestamps and source folder names in their names,
5. in addition to that error log files with error messages are produced and placed in the same Errors folders,
6. error log files are named in the following way: [Timestamp]\_[Processor Name]\_Error.log.

# System architecture

Please refer to [http://vcprdfdsmtfs03.royallondongroup.com/sites/ColA/IFDL/Acc/Shared%20Documents/Technical%20Documentation/BackOffice%20Tools%20-%20Technical%20documentation.docx](http://pg022wpapp004/sites/ColA/IFDL/Acc/Shared%20Documents/Technical%20Documentation/BackOffice%20Tools%20-%20Technical%20documentation.docx)

# User

Currently a particular user either has access to all 4 BOT pages with GUI or cannot access any of them. It is not possible to grant access to one page without granting the access to the remaining two pages.

# Features with GUI

## Bulk Novation

Bulk Novation is a page enabling Client Services Team to reassign multiple clients and accounts from one adviser to another. Such movements also include the changes of the marketing campaigns and corresponding expense groups. Successful bulk novation results in letters for the clients generated automatically.

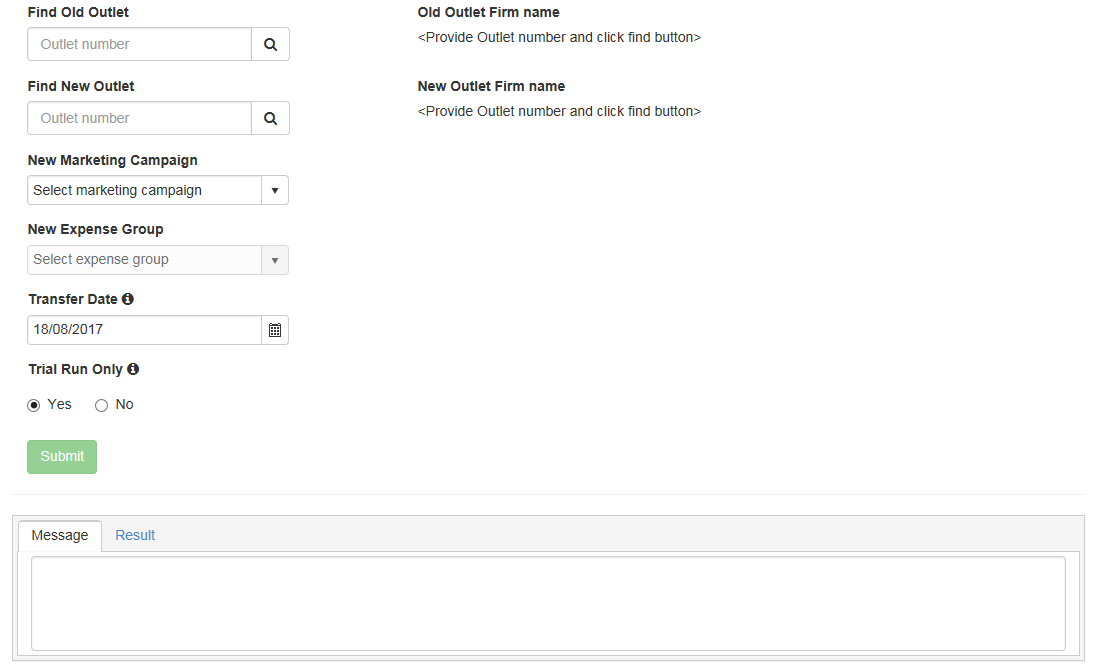
Please note that prior to performing bulk novation in Back Office Tools, the user needs to log into Sonata system and ensure that old and new marketing campaigns and expense groups are assigned to the appropriate advisers and clients/accounts (temporary configuration for the needs of bulk novation).

Please also note that after performing bulk novation in Back Office Tools:

* the user needs to log into Sonata system again and ensure that old and new marketing campaigns and expense groups are assigned to the appropriate advisers and clients/accounts (final configuration),
* the letters for the clients are generated automatically and then they need to be printed and sent,
* the letters for the advisers need to be generated manually.

On TSH environment the page is available at <http://tsh-backofficetools.platformservices.co.uk/BackOfficeTools/BulkNovation>

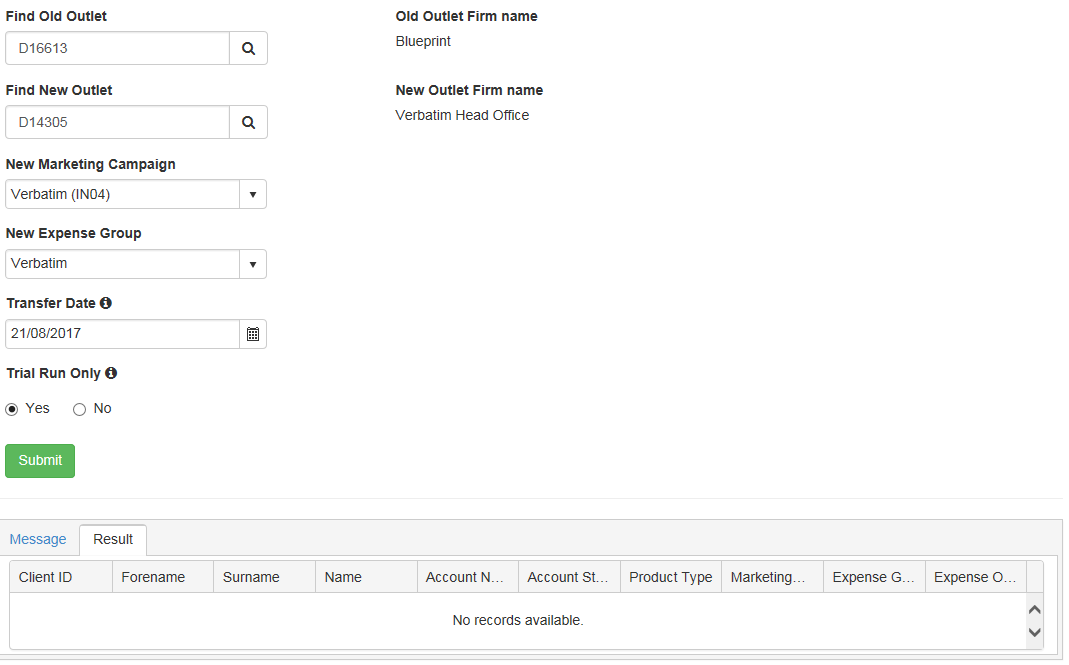
The screen shot below shows how the page looks.



The user needs to provide:

1. Old Outlet (please remember to click the magnifying glass icon),
2. New Outlet (please remember to click the magnifying glass icon),
3. New Marketing Campaign,
4. New Expense Group,
5. Transfer Date.

The screen shot below shows sample data entered.

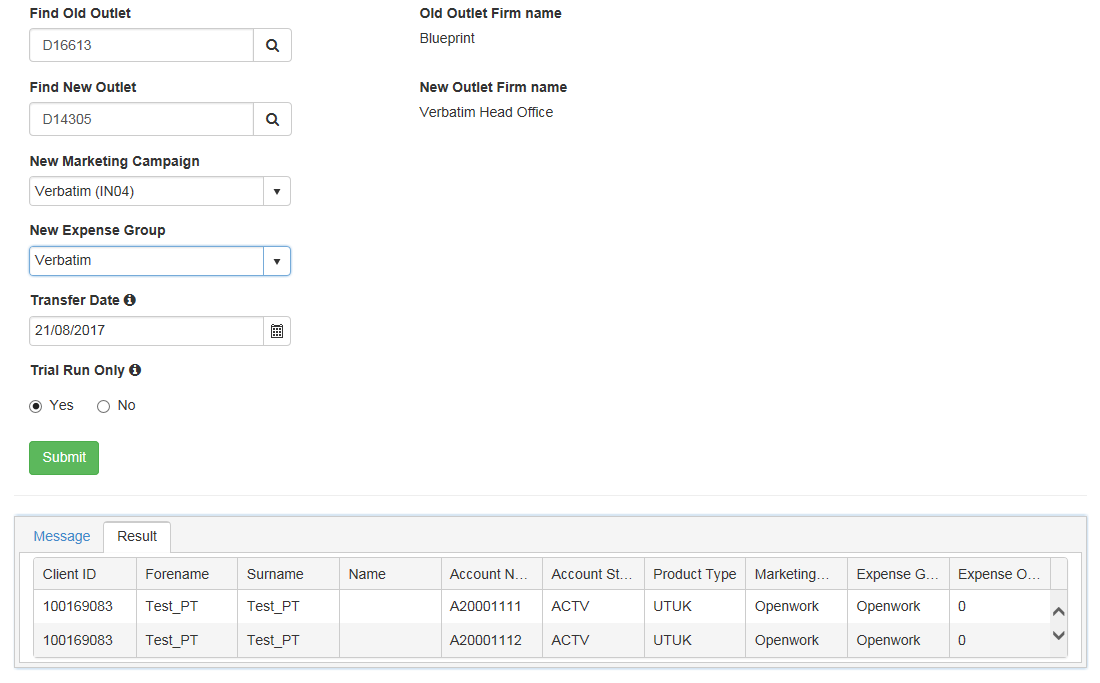


Then the user can choose between a trial run that does not change any assignments and a genuine bulk novation.

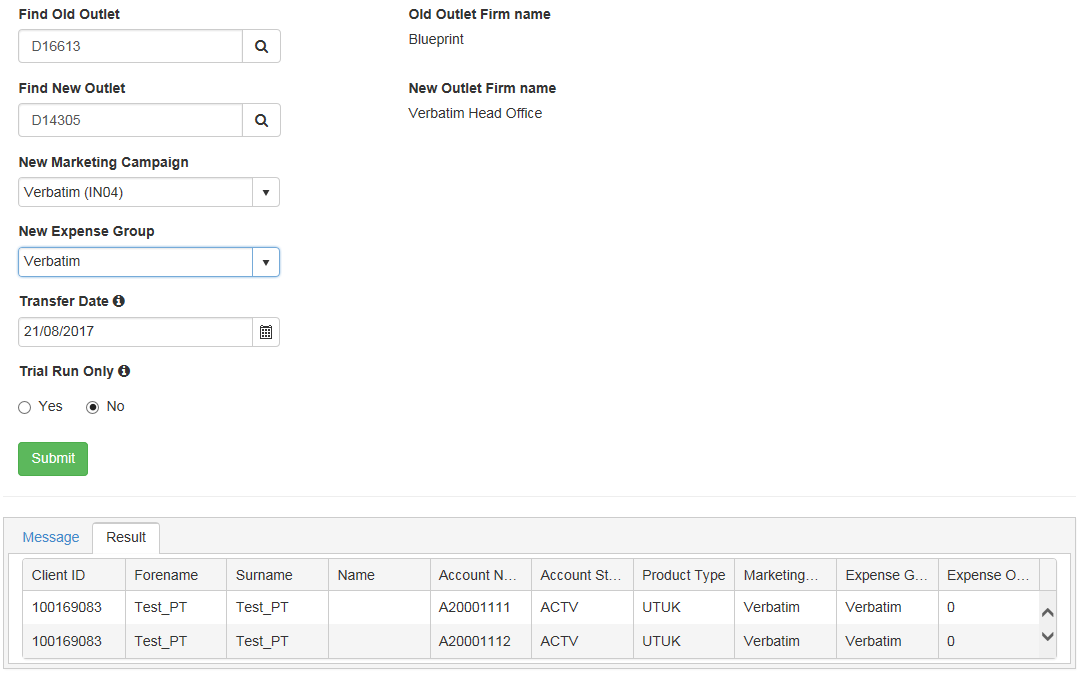
In order to start the execution, the user needs to click ‘Submit’ button.

Once the procedure has been run, the ‘Message’ tab at the bttom will contain the information whether or not the process has completed successfully whereas the ‘Result’ tab will contain a table with information about the accounts that have been moved.

The screen shot below presents the content of the ‘Result’ tab after a trial run. One can see that the Marketing Campaign and Expense Group have not been updated.

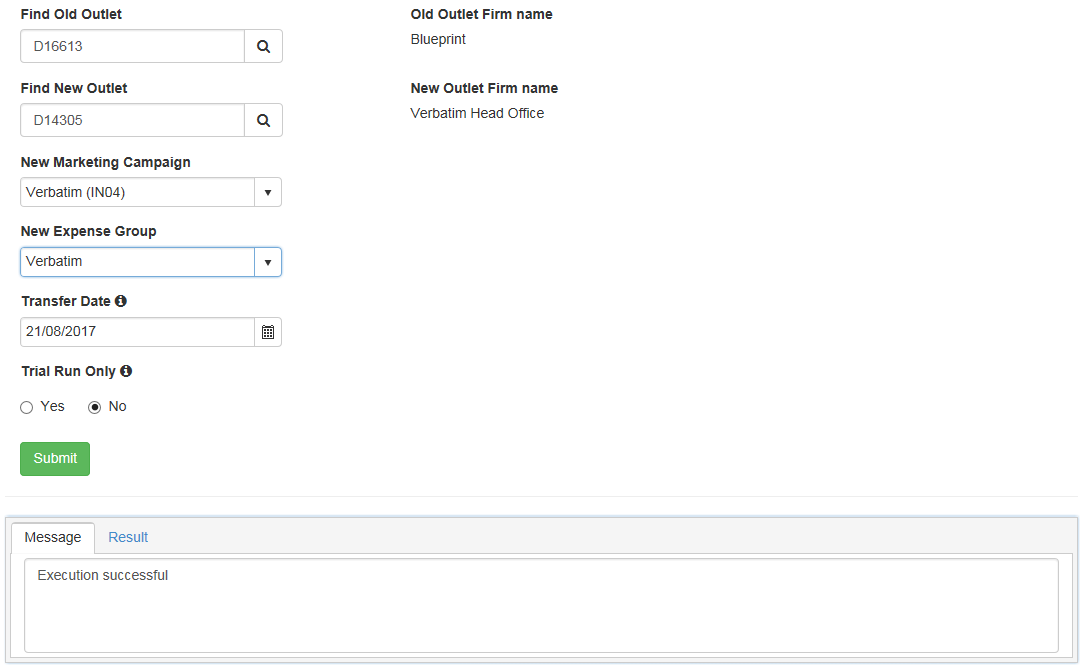


The screen shot below presents the content of the ‘Result’ tab after a genuine run. One can see that the Marketing Campaign and Expense Group have been updated.



The information can be copied from ‘Result’ tab to an Excel file for further processing (e.g. manual generation of the letters for the advisers).

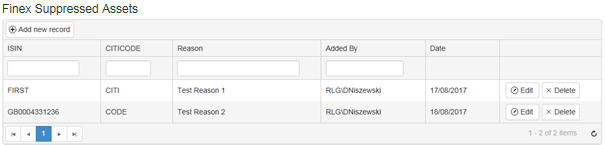
The screen shot below presents the content of the ‘Message’ tab after a genuine run. One can see that the execution has succeeded.



## ISIN Suppression for Finex asset prices

ISIN Suppression for Finex asset prices is a page enabling Fund and Stock Maintenance Team to maintain the list of assets to be excluded from Finex asset price files imported to Sonata. On TSH environment the page is available at <http://tsh-backofficetools.platformservices.co.uk/BackOfficeTools/finexsuppressionsmanagement>

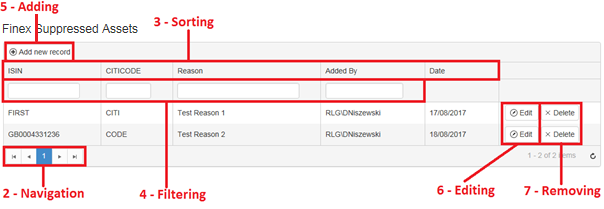
The screen shot below shows how the page looks.



The user can:

1. View the list of assets,
2. Navigate to a different page,
3. Sort the items on the list by a given column,
4. Filter the list by a given value in a selected column,
5. Add a new item to the list,
6. Edit the reason for an item on the list,
7. Remove an item from the list

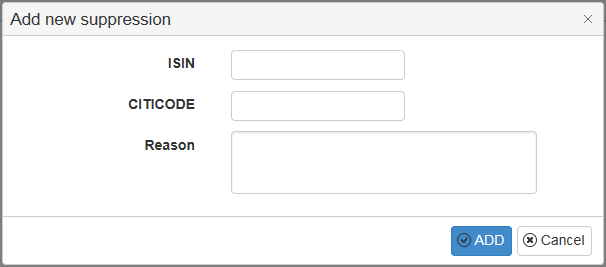
as depicted on the screen shot below.



In order to sort by a given column, the user needs to click on the header of that column. By default the list is not sorted. The first click results in acending order. The second click changes the order to descending whereas the third click removes the sorting.

In order to filter by a given value in a selected column, the user needs to type the value below the header in the appropriate column and press ‘Enter’.

In order to add an asset to the list, the user needs to click ‘Add new record’ button, type ISIN, CITICODE and Reason and click ‘ADD’ as depicted below.

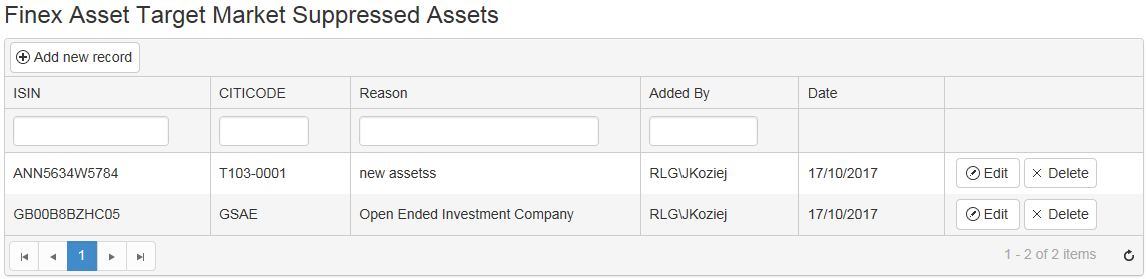


The suppression list is stored in table FinexIsinSuppression in BackOfficeInternal database.

## ISIN Suppression for Finex target market data

ISIN Suppression for Finex target market data is a page enabling Fund and Stock Maintenance Team to maintain the list of assets to be excluded from Finex target market data files imported to Sonata. On TSH environment the page is available at <http://tsh-backofficetools.platformservices.co.uk/BackOfficeTools/finexassettargetmarketsuppressionsmanagement>

The screen shot below shows how the page looks.



The user can:

1. View the list of assets,
2. Navigate to a different page,
3. Sort the items on the list by a given column,
4. Filter the list by a given value in a selected column,
5. Add a new item to the list,
6. Edit the reason for an item on the list,
7. Remove an item from the list

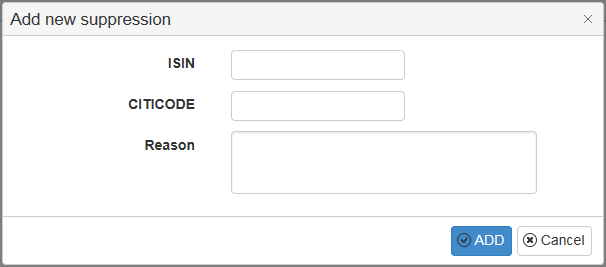
as depicted on the screen shot below.



In order to sort by a given column, the user needs to click on the header of that column. By default the list is not sorted. The first click results in acending order. The second click changes the order to descending whereas the third click removes the sorting.

In order to filter by a given value in a selected column, the user needs to type the value below the header in the appropriate column and press ‘Enter’.

In order to add an asset to the list, the user needs to click ‘Add new record’ button, type ISIN, CITICODE and Reason and click ‘ADD’ as depicted below.



The suppression list is stored in table FinexAssetTargetMarketIsinSuppresion in BackOfficeInternal database.

## Update Marketing Campaign for Adviser

Update Marketing Campaign for Adviser is a page enabling Client Services and Fees & Charges Teams to reassign multiple clients and accounts from one marketing campaign to another. Such movements also include the changes of the corresponding expense groups.

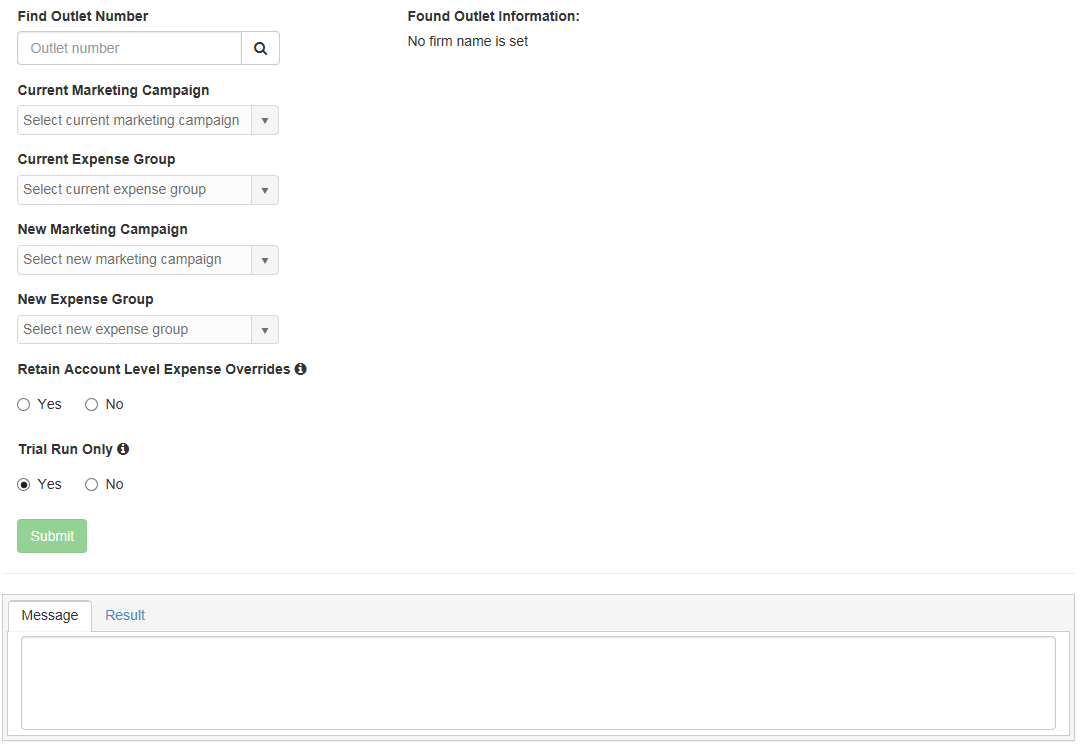
Please note that prior to performing the procedure in question in Back Office Tools, the user needs to log into Sonata system and ensure that old and new marketing campaigns and expense groups are assigned to the appropriate adviser and clients/accounts (temporary configuration for the needs of changing marketing campaign and expense group).

Please also note that after performing the procedure in question in Back Office Tools:

* the user needs to log into Sonata system again and ensure that old and new marketing campaigns and expense groups are assigned to the appropriate advisers and clients/accounts (final configuration),
* no letters are generated automatically.

On TSH environment the page is available at <http://tsh-backofficetools.platformservices.co.uk/BackOfficeTools/UpdateMarketingCampaignForAdvisor>

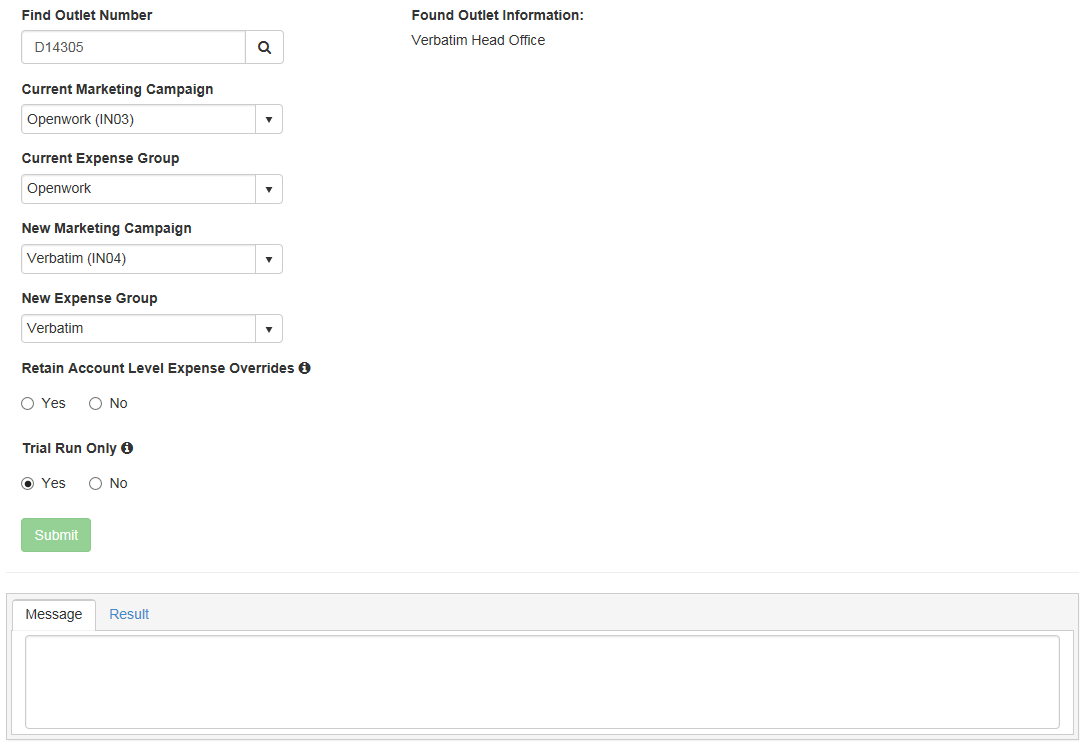
The screen shot below shows how the page looks.



The user needs to provide:

1. Outlet (please remember to click the magnifying glass icon),
2. Current Marketing Campaign,
3. Current Expense Group,
4. New Marketing Campaign,
5. New Expense Group,

The screen shot below shows sample data entered.



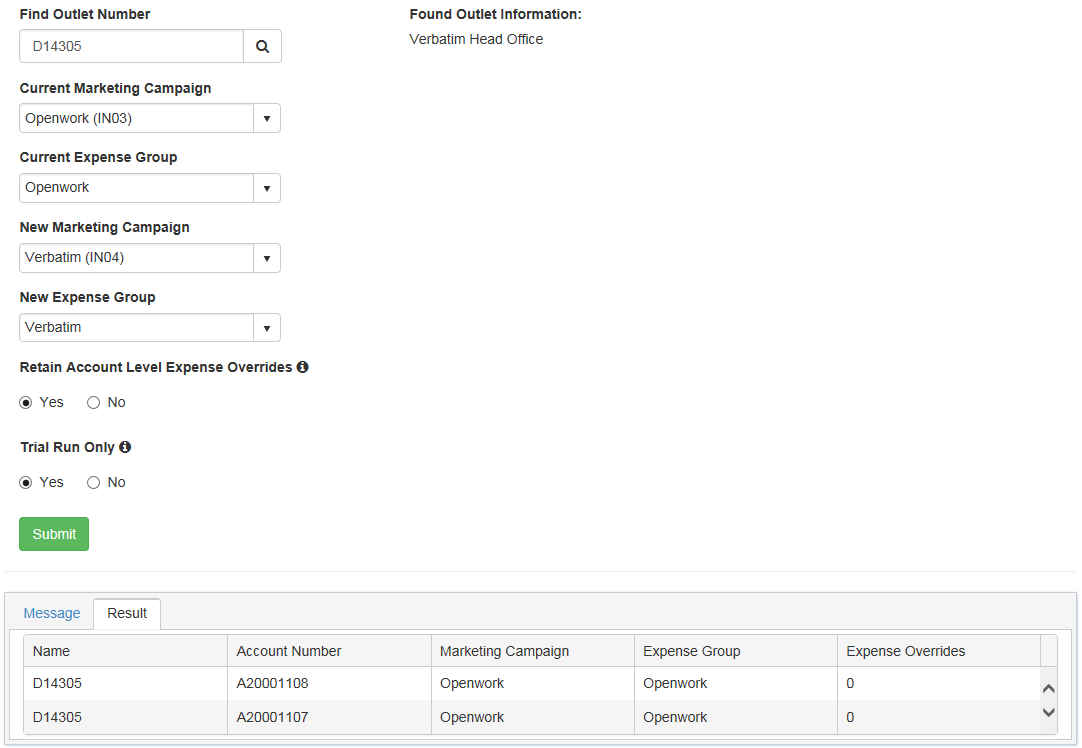
Then the user can decide whether or not the current fees set up for the account should remain the same.

Afterwards the user can choose between a trial run that does not change any assignments and a genuine update.

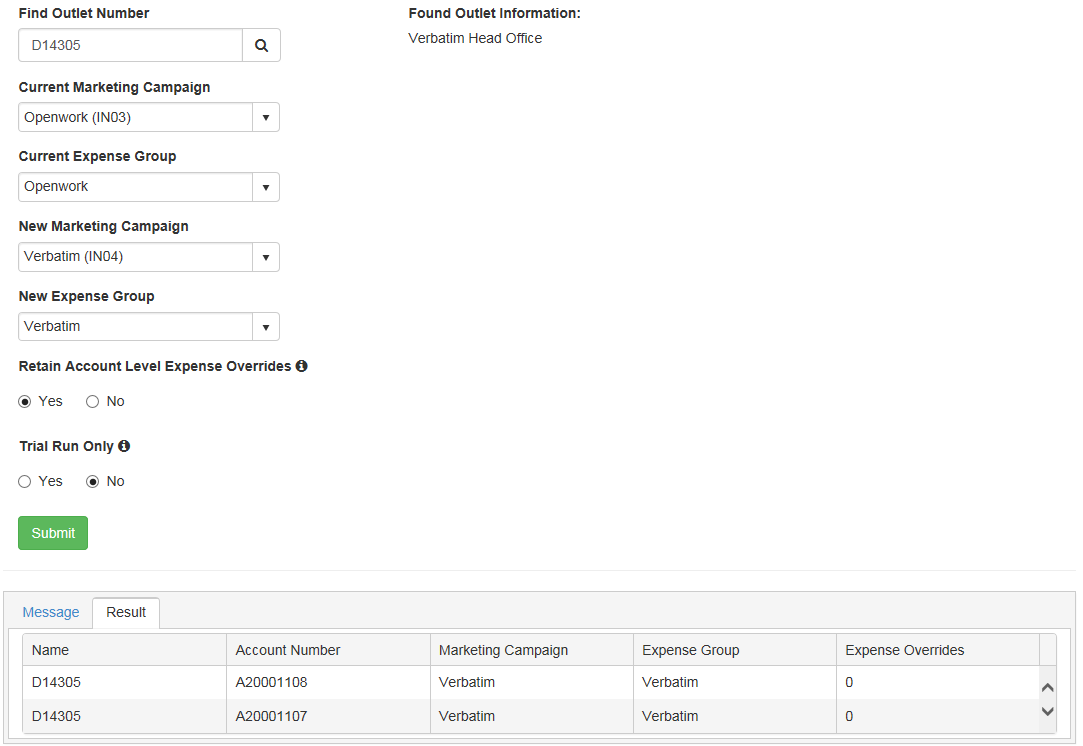
In order to start the execution, the user needs to click ‘Submit’ button.

Once the procedure has been run, the ‘Message’ tab at the bttom will contain the information whether or not the process has completed successfully whereas the ‘Result’ tab will contain a table with information about the accounts that have been impacted.

The screen shot below presents the content of the ‘Result’ tab after a trial run. One can see that the Marketing Campaign and Expense Group have not been updated.

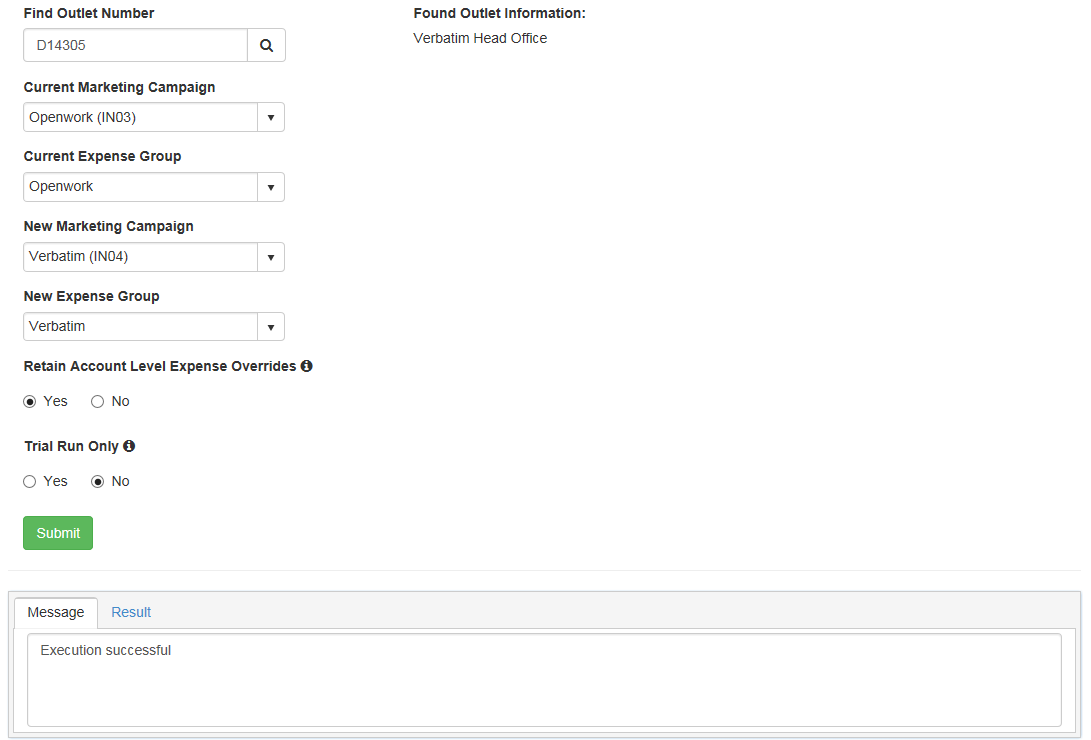


The screen shot below presents the content of the ‘Result’ tab after a genuine run. One can see that the Marketing Campaign and Expense Group have been updated.



The information can be copied from ‘Result’ tab to an Excel file if required.

The screen shot below presents the content of the ‘Message’ tab after a genuine run. One can see that the execution has succeeded.



# Watchers

## Altus Transfer Gateway

Altus processors wait for input Altus files containing information about account transfers. There are two processors. The former waits for the so called confirmed files and the latter waits for the so called verified files. Once the confirmed file has been detected (it needs to contain ‘Confirmed’ in the filename), the processor splits it into 2 output files:

* ‘Confirmed Transfer In Assets Date/timestamp from Altus POST BOT to Sonata.csv’ for Sonata,
* ‘Confirmed Transfer In Assets Date/timestamp from Altus POST BOT to BB.csv’ for Blue Button.

Once the verified file has been detected (it needs to contain ‘Verified’ in the filename), the processor splits it into 2 output files:

* ‘Verified Date/timestamp from Altus POST BOT to Sonata.csv’ for Sonata,
* ‘Verified Date/timestamp from Altus POST BOT to BB.csv’ for Blue Button.

While producing the output files, BOT amends the values in one of the columns by removing ‘AC’ (denoting Sonata) and ‘BB’ (denoting Blue Button) prefixes. The values in all the other columns remain unchanged.

The folder structure for confirmed files is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | Altus |  |  |  |  |
|  | L |  |  |  |  |
|  |  | Archive |  |  | Archive |
|  |  | Errors |  |  | Files in error and error log |
|  |  | Files |  |  |  |
|  |  | L |  |  |  |
|  |  |  | FromAltus |  |  |
|  |  |  | L | Confirmed | Input folder |
|  |  |  | ToBB |  | Output for Blue Button |
|  |  |  | ToSonata |  | Output for Sonata |

The folder structure for verified files is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | Altus |  |  |  |  |
|  | L |  |  |  |  |
|  |  | Archive |  |  | Archive |
|  |  | Errors |  |  | Files in error and error log |
|  |  | Files |  |  |  |
|  |  | L |  |  |  |
|  |  |  | FromAltus |  |  |
|  |  |  | L | Verified | Input folder |
|  |  |  | ToBB |  | Output for Blue Button |
|  |  |  | ToSonata |  | Output for Sonata |

## Asset Classification

Asset Classification processor waits for an input Excel file filled in manually by the users containing asset classification data. Please note that asset classification data needs to be uploaded to Sonata provided that the basic asset data is already there. The basic asset data can be prepared using Asset Loader BOT processor. Once the asset classification file has been detected, the processor pulls the data from the ‘Asset Classification’ data sheet, validates it by checking ISINs against ODS database and performing additional checks, enriches it with some additional information and outputs the result into a CSV file that can be imported by Sonata system.

The output file contains the following data.

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Comment** |
| Record ID | Unique Identifier | Autoincrement |
| Asset Name | ISIN | From column A in the spreadsheet |
| <> | Leave Blank – Other columns used to identify Asset |  |
| <> | Leave Blank – Other columns used to identify Asset |  |
| Classification Type | Sonata Type Code CLAT code | From column B in the spreadsheet |
| Source | Sonata Type Code CLSO code | From column D in the spreadsheet |
| Description | Free Text (50 Char) | From column E in the spreadsheet |
| Comment | Free Text (50 Char) | User name (Last Modified Property in input Excel spreadsheet) + today’s date |

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | BackOfficeTools |  |  |  |  |
|  | L | AssetClassification |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromStockMaintenance | Input folder |
|  |  |  |  | ToSonata | Output folder |

## Asset Loader

Asset Loader processor waits for an input Excel file filled in manually by the users containing basic data for assets of a particular type. Once the file has been detected, the processor pulls the data from the create and modify data sheets, combining them into one sheet and outputs it as CSV so that the output file produced can be imported by Sonata system.

Sample Excel input file is embedded below.



The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | BackOfficeTools |  |  |  |  |
|  | L | AssetLoader |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromStockMaintenance | Input folder |
|  |  |  |  | ToSonata | Output folder |

## Bloomberg

Bloomberg processor waits for an input Bloomberg file containing asset prices. Once the file has been detected, the processor changes the format of the file and performs some calculations so that the output file produced can be imported by Sonata system.

The format changes and calculations include:

* file name,
* file extension (csv),
* the content of the file (values from the input file, hardcoded values and values calculated based on the values from the input file and from ODS database).

All errors found are reported at the end of the process, rather than fail-on-first.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |
| L | Bloomberg |  |  |  |
|  | L |  |  |  |
|  |  | Archive |  | Archive |
|  |  | Errors |  | Files in error and error log |
|  |  | Files |  |  |
|  |  | L |  |  |
|  |  |  | FromBloomberg | Download from Bloomberg – input for BOT processing |
|  |  |  | ToSonata | Files for Sonata – output from BOT processing |

## Calastone

There are two Calastone processors. The former waits for an input Sonata file containing orders whereas the latter waits for a confirmation input file from Calastone. Once the Sonata orders file has been detected, the processor changes its format and content so that the output file produced can be imported by Calastone system. File content transformations include, among others, adding information from ODS database.

The folder structure for the first processor is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L |  |  |  |  |  |
|  | Calastone |  |  |  |  |
|  | L |  |  |  |  |
|  |  | Archive |  |  | Archive |
|  |  | Errors |  |  | Files in error and error log |
|  |  | Files |  |  |  |
|  |  | L | ToCalastone |  | Files for Calastone – output from BOT |
|  | Sonata |  |  |  |  |
|  | L | Calastone |  |  |  |
|  |  | L | Files |  |  |
|  |  |  | L | ToCalastone | Orders file from Sonata – input for BOT processing |

Once the confirmation file from Calastone has been detected, the processor changes its format and content so that the output file produced can be imported by Sonata system.

The folder structure for the second processor is as follows. The example is for TSH environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |
| L | Calastone |  |  |  |
|  | L |  |  |  |
|  |  | Archive |  | Archive |
|  |  | Errors |  | Files in error and error log |
|  |  | Files |  |  |
|  |  | L |  |  |
|  |  |  | FromCalastone | Confirmation files from Calastone – input for BOT processing |
|  |  |  | ToSonata | Files for Sonata – output from BOT |

## Distribution FINEX

Distribution FINEX processor waits for an input Finex file containing information about dividends related to assets. Once the file has been detected, the processor splits it into 3 output files:

* FinexDistribution\_IsinWithHolder.csv – contains data for dividends for assets with holders,
* FinexDistribution\_IsinWithoutHolder.csv – contains data for dividends for assets with no holders,
* FinexDistribution\_IsinNotFound.csv – contains data for assets where ISINs are not recognized.

The last output file is not imported to Sonata whereas the first two are.

All 3 output files are not always produced. Each file is generated only if there is at least one asset that qualifies for it.

The columns in the second output file differ from the columns in the input file whereas the columns in the first and the third output files are the same as the columns in the input file. While producing the second output file, BOT takes some values from the input file and performs some calculations.

Further details can be found in [BackOfficeTools-Distribution\_Finex Manual\_V4](http://pg022wpapp004/sites/ColA/IFDL/Acc/Shared%20Documents/Objectivity/Documentation/User%20Guides/BackOfficeTools-Distribution_Finex%20Manual_V4.docx)

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |
| L | Finex |  |  |  |
|  | L |  |  |  |
|  |  | Archive |  |  |
|  |  | L | FinexDist | Archive |
|  |  | Errors |  |  |
|  |  | L | FinexDist | Files in error and error log |
|  |  | Files |  |  |
|  |  | L |  |  |
|  |  |  | FromFinexDist | Input folder |
|  |  |  | ToSonata | Output folder |

## EMX Fund List

EMX Fund List processor waits for an input EMX file containing information about fund asset statuses. Once the file has been detected, the processor changes its format and content so that the output file can be imported by Sonata system allowing Sonata to create valid OINP EMX messages.

The format changes and calculations include:

* file name,
* file extension (csv) – the output file is ‘EMXFundList.csv’
* the content of the file (values from the input file, hardcoded values and values calculated based on the values from the input file and from ODS database, rows filtered out based on the values from the input file and from ODS database).

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |
| L | EMX |  |  |  |
|  | L |  |  |  |
|  |  | Archive |  | Archive |
|  |  | Errors |  | Files in error and error log |
|  |  | Files |  |  |
|  |  | L |  |  |
|  |  |  | FromEMX | Download from EMX – input for BOT processing |
|  |  |  | ToSonata | Files for Sonata – output from BOT processing |

## Finex asset prices

Finex asset prices processor waits for an input Finex file containing asset prices. There are two types of files: equity price and fund price. Once the file has been detected, the processor changes the format of the file and performs some calculations so that the output file produced can be imported by Sonata system.

The format changes and calculations include:

* file name,
* file extension for equity price file (csv) – the output files are ‘FinexAssetPrices\_Equity.csv’ and ‘FinexAssetPrices\_Fund.csv’,
* the content of the file (values from the input file, hardcoded values and values calculated based on the values from the input file and from ODS database, rows filtered out (due to exclusions/errors) based on the values from the input file and from ODS database).

All errors/exclusions found are reported at the end of the process, rather than fail-on-first.

The following categories of exclusions are defined:

* Unknown,
* Asset not setup in Sonata,
* Price suppressed,
* Blank ISIN Code,
* Duplicate foreign currency price,
* Foreign currency price with home currency,
* Duplicate home currency price on same date,
* Lower ranked foreign currency price,
* Home currency price for an older date,
* Price already received for date.

In case exclusion ‘Duplicate home currency price on same date’ is detected, then BOT passes appropriate information to Notification Hub and then Notification Hub generates the following email and sends it to Stock & Funds Maintenance Team:

Email subject:

Finex Duplicate Price Error for Investigation in file {filename}

({} indicates dynamic fields)

Example: Finex Duplicate Price Error for Investigation in file 20170531\_fdxv2201.csv

Email body:

ISIN {ISIN code}, Citicode {Citicode} duplicate {Original currency} price found for {Price date} in {equity/fund} file {filename}:

Bid: {Bid price}

Offer: {Offer price}

Mid: {Mid price}

({} indicates dynamic fields)

NOTE 1:

The filename is the original Finex name

NOTE 2:

All exclusions of the category in question related to one file to be sent in one message

NOTE 3:

The ISINs to be seperated by additional line

Example:

ISIN GB0004198445, Citicode 06HB duplicate GBP price found for 31/05/2017 in fund file 20170531115413380\_fdxv220170531.csv:

Bid: 0.5276

Offer: 0.5284

Mid: 0

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |
| L | Finex |  |  |  |
|  | L |  |  |  |
|  |  | Archive |  |  |
|  |  | L | Finex | Archive |
|  |  | Errors |  |  |
|  |  | L | Finex | Files in error and error/exclusion log |
|  |  | Files |  |  |
|  |  | L |  |  |
|  |  |  | FromFinex | Input folder |
|  |  |  | ToSonata | Output folder |

## Finex target market data

Finex target market data processor waits for an input Finex file containing asset target market data. The input file name is ‘FINEX-MIFID-V3-FULL-yyyymmdd.csv’. Once the file has been detected, the processor changes the format of the file and performs some calculations so that the output file produced can be imported by Sonata system.

The format changes and calculations include:

* file name – the output file is ‘AssetTargetMarketFull.csv’,
* the content of the file (values from the input file, hardcoded values and values calculated based on the values from the input file and from ODS database, rows filtered out (due to exclusions/errors and due to no difference against ODS database) based on the values from the input file and from ODS database).

All errors/exclusions found are reported at the end of the process, rather than fail-on-first.

The following categories of exclusions are defined:

* Missing ISIN,
* ISIN outside of Ascentric universe,
* Duplicated ISIN,
* Suppressed ISIN,
* Mandatory fields are blank,
* Some fields with incorrect values.

Rows with no difference against ODS database are also excluded, but they are not stored in the exclusion file.

Maping and transformation file is embedded below.



File with process steps is embedded below.



The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | Finex |  |  |  |  |
|  | L |  |  |  |  |
|  |  | Archive |  |  |  |
|  |  | L | AssetTargetMarket |  | Archive |
|  |  | Errors |  |  |  |
|  |  | L | AssetTargetMarket |  | Files in error and error/exclusion log |
|  |  | Files |  |  |  |
|  |  | L |  |  |  |
|  |  |  | FromFinexAssetTargetMarket |  |  |
|  |  |  | L | Full | Input folder |
|  |  |  | ToSonata |  | Output folder |

## GAD and SMPI/ARS

GAD processor waits for an input Dunstan Thomas file containing illustration of pension prediction. Once the file has been detected, the processor extracts client id from the file name, starts letter generation process (Sonata Gen Letter service is called, letter T178) and moves the file to two folders. The former is the standard output folder and the latter is the auxiliary output folder. The file placed in the standard output folder is not used whereas the file placed in the auxiliary output folder is attached to the letter.

The rationale behind having two output folders is as follows:

From the Business point of view, all BOT processors except for GAD and SMPI/ARS require that when BOT creates an output file and a file with identical name already exists in the output folder, then ‘\_1’ suffix is appended to the new output file name instead of overwriting the old file with the new one. BOT can be globally configured either to add suffixes or to overwrite files in case of name conflict. It is not possible to configure each processor separately and independently. That is why BOT is globally configured to add suffixes, which is not in line with the requirements for GAD and SMPI/ARS. To solve this issue, GAD and SMPI/ARS processors output the files not only to the standard output folders where suffixes are added in case of name conflicts, but also to the auxiliary ones where files are overwritten in case of name conflicts.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |  |
| L | DunstanThomas |  |  |  |  |  |
|  | L |  |  |  |  |  |
|  |  | Archive |  |  |  |  |
|  |  | L | GAD |  |  | Archive |
|  |  | Errors |  |  |  |  |
|  |  | L | GAD |  |  | Files in error and error log |
|  |  | Files |  |  |  |  |
|  |  | L | ToPrintService |  |  |  |
|  |  |  | L | GAD |  |  |
|  |  |  |  | L |  |  |
|  |  |  |  |  | In | Input folder |
|  |  |  |  |  | Out | Standard output folder |
|  | PrintService |  |  |  |  |  |
|  | L | Bulk |  |  |  |  |
|  |  | L | Enclosures |  |  |  |
|  |  |  | L | EDRE |  | Auxiliary output folder |

SMPI/ARS processor works in an analogous way. Letter type (T026a) and folders are different in this case.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |  |
| L | DunstanThomas |  |  |  |  |  |
|  | L |  |  |  |  |  |
|  |  | Archive |  |  |  |  |
|  |  | L | SMAL |  |  | Archive |
|  |  | Errors |  |  |  |  |
|  |  | L | SMAL |  |  | Files in error and error log |
|  |  | Files |  |  |  |  |
|  |  | L | ToPrintService |  |  |  |
|  |  |  | L | SMAL |  |  |
|  |  |  |  | L |  |  |
|  |  |  |  |  | In | Input folder |
|  |  |  |  |  | Out | Standard output folder |
|  | PrintService |  |  |  |  |  |
|  | L | Bulk |  |  |  |  |
|  |  | L | Enclosures |  |  |  |
|  |  |  | L | SMAL |  | Auxiliary output folder |

## Generic Data Extract

For some time Blue Button and Sonata systems work on production simultaneously – this is referred to as dual running. According to the initial plan as a first step (Step 1), data related to HSBC Onshore Bond (administered by Opal) Omnibus accounts was to be migrated to Sonata. As a second step (Step 2), data related to HSBC non-Omnibus accounts, Prudential and Scottish Friendly was to be migrated to Sonata.

However, an alternative approach assuming that all data (HSBC Omnibus accounts, HSBC non-Omnibus accounts, Prudential and Scottish Friendly) will be migrated within a single step has been chosen. This is equivalent to Step 2 in the initial plan as far as BOT configuration is concerned.

Before migration HSBC/Opal, Prudential and Scottish Friendly prepared inbound files for Blue Button. After the migration, when dual running starts, Back Office Tools (BOT) has to split these files into Sonata part and Blue Button part.

Besides, before migration Blue Button prepared outbound files for HSBC/Opal, Prudential and Scottish Friendly. After the migration, when dual running starts, both Blue Button and Sonata prepare such files. BOT has to merge these files so that dual running is transparent for HSBC/Opal, Prudential and Scottish Friendly.

There are 3 inbound (split) processors for HSBC/Opal, Prudential and Scottish Friendly and 3 outbound (merge) processors for HSBC/Opal, Prudential and Scottish Friendly.

As far as inbound processors are concerned, BOT expects only the following files in the input folder:

* 1 file for HSBC Omnibus Accounts – Step 1 configuration (FCDDMMYY) – not used,
* 2 files for HSBC – Step 2 configuration (FCDDMMYY and FNDDMMYY),
* 2 files for Pru – Step 2 configuration (CASHTXNS\_YYYYMMDD and PIA\_New\_Policies\_YYYYMMDD),
* 2 files for SF – Step 2 configuration (FCDDMMYY and FNDDMMYY).

BOT splitting for HSBC/Opal will start only if all of the expected files for HSBC/Opal have arrived. Otherwise the BOT splitting for HSBC/Opal will not start at all. The same applies for Prudential and Scottish Friendly.

All the input and output files mentioned above will always be produced (can be empty though).

Both Sonata and BB files retain the original name and format.

The Sonata files contain the rows related to accounts live on Sonata whereas the other rows are in BB files.

Step 1 configuration (not used) uses hardcoded solution to recognize the rows related to accounts live on Sonata whereas Step 2 configuration uses information from ODS and RIMA databases for that purpose.

As far as outbound processors are concerned, BOT will expect only the following files in the input folder:

* **21 files for HSBC – Step 1 configuration – 12 from BB and 9 from Sonata – not used** (INTEREST\_YYYYMMDD (BB only), SETTCASHTXNS\_YYYYMMDD (BB only), SETTUNITTXNS\_YYYYMMDD (BB only), CASHTXNS\_YYYYMMDD, UNITTXNS\_YYYYMMDD, DISTS\_YYYYMMDD, FIN\_TRANS\_YYYYMMDD, HSBC\_PAYMENT\_YYYYMMDD, UNIT\_ASSETS\_YYYYMMDD, CASH\_ASSETS\_YYYYMMDD, FIN\_DISTRIBUTION\_YYYYMMDD, FIN\_HOLDINGS\_YYYYMMDD),
* **22 files for HSBC – Step 2 configuration – 12 from BB and 10 from Sonata** (SETTCASHTXNS\_YYYYMMDD (BB only), SETTUNITTXNS\_YYYYMMDD (BB only), CASHTXNS\_YYYYMMDD, UNITTXNS\_YYYYMMDD, DISTS\_YYYYMMDD, FIN\_TRANS\_YYYYMMDD, OPAL\_New\_Policies\_YYYYMMDD, HSBC\_PAYMENT\_YYYYMMDD, UNIT\_ASSETS\_YYYYMMDD, CASH\_ASSETS\_YYYYMMDD, FIN\_DISTRIBUTION\_YYYYMMDD, FIN\_HOLDINGS\_YYYYMMDD),
* **18 files for Pru – Step 2 configuration – 9 from BB and 9 from Sonata** (CASHTXNS\_YYYYMMDD, UNITTXNS\_YYYYMMDD, DISTS\_YYYYMMDD, FIN\_TRANS\_YYYYMMDD, PIA\_New\_Policies\_YYYYMMDD, UNIT\_ASSETS\_YYYYMMDD, CASH\_ASSETS\_YYYYMMDD, FIN\_DISTRIBUTION\_YYYYMMDD, FIN\_HOLDINGS\_YYYYMMDD)
* **22 files for SF – Step 2 configuration – 12 from BB and 10 from Sonata** (SETTCASHTXNS\_YYYYMMDD (BB only), SETTUNITTXNS\_YYYYMMDD (BB only), CASHTXNS\_YYYYMMDD, UNITTXNS\_YYYYMMDD, DISTS\_YYYYMMDD, FIN\_TRANS\_YYYYMMDD, SF\_New\_Policies\_YYYYMMDD,SF\_PAYMENT\_YYYYMMDD, UNIT\_ASSETS\_YYYYMMDD, CASH\_ASSETS\_YYYYMMDD, FIN\_DISTRIBUTION\_YYYYMMDD, FIN\_HOLDINGS\_YYYYMMDD).

BOT merging processing for HSBC/Opal will start only if all of the expected files for HSBC/Opal have arrived. Otherwise the BOT merging processing for HSBC/Opal will not start at all.

The same applies for Prudential and Scottish Friendly.

Besides, as far as the files that contain date in the name are concerned, the BB file and the corresponding Sonata file need to have the same date. Otherwise the BOT merging processing will not start at all.

While merging the files, BOT will apply the following rules:

1. If both files exist, but the Sonata file is empty, then ignore the Sonata file while processing,
2. If both files exist, but the BB file is empty, then ignore the BB file while processing,
3. If both files exist and both are empty, then ignore the BB file while processing.

The output files retain the original name and format with the exception of payment files where the files are not merged and receive ‘\_BB’ and ‘\_Sonata’ suffixes instead.

The transformations during merging include:

* appending the content of one file at the end of the other one,
* removing the rows related to accounts live on Sonata from the BB file,
* groupings and calculations,
* removing the rows with transaction code T101 from the BB file (the list of transaction codes configurable in the database).

Please note that that transformations are not the same for each pair of files.

Step 1 configuration (not used) uses hardcoded solution to recognize the rows related to accounts live on Sonata whereas Step 2 configuration uses information from ODS and RIMA databases for that purpose.

BB FIN\_DISTRIBUTION\_YYYYMMDD files may contain lines with ISIN that does not appear in corresponding Sonata files and vice versa.

In either case such lines should be taken into account while generating the output files (included in the output files) and no errors should be flagged.

While summing up values during grouping, BOT should take the input values ‘as is’ (no matter how many decimal places they have) and then it should calculate the aggregated (summed up) output value without removing any decimal places.

BB FIN\_HOLDINGS\_YYYYMMDD files may contain lines with a combination of ISIN and Product Short Code that does not appear in corresponding Sonata files and vice versa.

In either case such lines should be taken into account while generating the output files (included in the output files) and no errors should be flagged.

For ISIN existing both in BB and Sonata file, the price from the Sonata file should be taken.

For ISIN existing only in BB file, the price from the BB file should be taken.

For ISIN existing only in Sonata file, the price from the Sonata file should be taken.

It is assumed that in a given system the price for a particular ISIN is the same, no matter what the ‘Product Short Code’ is. BOT will create the first list of ISINs and their prices based on Sonata file only and the second list of ISINs and their prices based on BB file only. While grouping the merged content of both files by ISIN by ‘Product Short Code’, for each ISIN the price from the Sonata price list will be taken. If the ISIN is not there, the price from the BB price list will be taken. Thus, Sonata price will be used whenever possible. Otherwise BB price will be used.

While summing up values during grouping, BOT should take the input values ‘as is’ (no matter how many decimal places they have) and then it should calculate the aggregated (summed up) output value without removing any decimal places – please note that the output value for column G (aggregated Quantity (column C) \* Price (column F)) will be rounded down to 2 decimal places though.

Further details can be found in [Dual\_Running\_HSBC\_non\_Omnibus\_Prudential\_Scottish\_Friendly\_with\_Inflights\_BOT](http://pg022wpapp004/sites/ColA/IFDL/Acc/Shared%20Documents/Dual_Running_HSBC_non_Omnibus_Prudential_Scottish_Friendly_with_Inflights_BOT.docx)

The folder structure is as follows. The example is for TSH environment.

* for HSBC/Opal (both inbound and outbound processors)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | ODSExtracts |  |  |  |  |
|  | L | HSBC |  |  |  |
|  |  | L | Archive |  |  |
|  |  | L | Errors |  |  |
|  |  | L | Files |  |  |
|  |  |  | L | FromHSBCPreSplit | ftp download from HSBC/Opal – input for BOT split processing |
|  |  |  | L | FromBB | input for BOT from BB extract |
|  |  |  | L | FromHSBC | output for Sonata from BOT split processing |
|  |  |  | L | FromSonata | output from Sonata for BOT merge processing |
|  |  |  | L | ToBB | output for BB from BOT split processing |
|  |  |  | L | ToHSBC | output from BOT merge processing for file going back to HSBC/Opal |
|  |  |  | L | ToSonata | files for Sonata (not used by BOT) |

* for Prudential (both inbound and outbound processors)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | ODSExtracts |  |  |  |  |
|  | L | Prudential |  |  |  |
|  |  | L | Archive |  |  |
|  |  | L | Errors |  |  |
|  |  | L | Files |  |  |
|  |  |  | L | FromPrudentialPreSplit | ftp download from Prudential – input for BOT split processing |
|  |  |  | L | FromBB | input for BOT from BB extract |
|  |  |  | L | FromPrudential | output for Sonata from BOT split processing |
|  |  |  | L | FromSonata | output from Sonata for BOT merge processing |
|  |  |  | L | ToBB | output for BB from BOT split processing |
|  |  |  | L | ToPrudential | output from BOT merge processing for file going back to Prudential |
|  |  |  | L | ToSonata | files for Sonata (not used by BOT) |

* for Scottish Friendly (both inbound and outbound processors)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | ODSExtracts |  |  |  |  |
|  | L | SF |  |  |  |
|  |  | L | Archive |  |  |
|  |  | L | Errors |  |  |
|  |  | L | Files |  |  |
|  |  |  | L | FromSFPreSplit | ftp download from Scottish Friendly – input for BOT split processing |
|  |  |  | L | FromBB | input for BOT from BB extract |
|  |  |  | L | FromSF | output for Sonata from BOT split processing |
|  |  |  | L | FromSonata | output from Sonata for BOT merge processing |
|  |  |  | L | ToBB | output for BB from BOT split processing |
|  |  |  | L | ToSF | output from BOT merge processing for file going back to Scottish Friendly |
|  |  |  | L | ToSonata | files for Sonata (not used by BOT) |

## HSBC ACK Processing

When HSBC receives Sonata payment files from IFDL, they are validated immediately and an acknowledgement file is returned to IFDL (referred to as an “ACK”). These files contain either confirmation that the file and payments were valid, or a description of any errors found. In either case, the IFDL Payments Team needs to be aware of the acknowledgements and act on them as required.

The level of detail in the acknowledgement files does not make it easy to identify the file being acknowledged, a specific payment being referred to, or the nature of any error found. Therefore, BOT extracts and stores information from payment files, compiles it with the information from ACK files and then generates meaningful messages that are sent via email to the Payment Team.

There are two types of payment files – BACS and MT103 (CHAPS and Faster Payments). The former triggers generation of a single ACK file per 1 payment file whereas the latter generation of 1 or 2 ACK files per 1 payment file, depending on the kinds of errors found.

Payment file processor waits for an input payment file. Once the file has been detected, the processor extracts certain data, stores it in the database and moves the file to the output folder so that it can be delivered to HSBC.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | Sonata |  |  |  |  |
|  | L | HSBC |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromSonata | Input folder |
|  |  |  |  | ToHSBC | Output folder |

ACK file processor waits for an input ACK file. Once the file has been detected, the processor extracts certain data, waits for the other ACK file if needed, tries to match the ACK file/s with the corresponding payment file, passes appropriate information to Notification Hub and then Notification Hub generates the following emails and sends them to the Payments Team.

**Email 1** - generated once for each payment file that has been acknowleged by HSBC (there is file reference inside the ACK file/s and BOT can find the matching payment file)

Email subject:

Payment file <File name> <CNTL error message> <Paymentstatus>

(<> indicates dynamic fields)

Example: Payment file IFDL\_HSBC\_20161028\_161032.TXT Valid file with payment error

Email body:

<File type> file <File name> <CNTL error message>.

<Totalnum> valid payments of <Currency> <Totalccy>.

Error payment number <Number>, reference <Ref>, destination <Bank> <AccountNumber>, name <AccountName>, <Amount> <Ccy>, reason: <BANSTA error message>.

Summary: <Totalvalid> valid payments and <Totalinvalid> invalid payments.

Example:

BACS file IFDL\_HSBC\_20161028\_161032.TXT Valid file.

4 valid payments of GBP 2,900.99.

Error payment number 3, reference A20000149 AB564673, destination 404786 10284408, name Mr Test ACK3, 700.90 GBP, reason: INVALID DEBIT ACCOUNT.

Summary: 4 valid payments and 2 invalid payments.

**Email 2** - generated once for each ACK file that cannot be matched with appropriate payment file (BOT cannot read ACK file (file empty, corrupt, invalid format, etc.))

Email subject:

Cannot read ACK file <ACK file name>

(<> indicates dynamic fields)

Example: Cannot read ACK file LCDE.BLQSAM.OCABI481.C0168533.b7e18l04

Email body:

Cannot read ACK file <ACK file name>.

Example: Cannot read ACK file LCDE.BLQSAM.OCABI481.C0168533.b7e18l04.

**Email 3** - generated once for each ACK file that cannot be matched with appropriate payment file (there is file reference inside the ACK file/s, but despite this BOT is unable to find the matching payment file)

Email subject:

Cannot find matching payment file

Example: Cannot find matching payment file

Email body:

Cannot find matching payment file for ACK file <ACK file name>.

Example:

Cannot find matching payment file for ACK file LCDE.BLQSAM.OCABI481.C0168533.b7e18l04.

**Email 4** - generated once for each ACK file that cannot be matched with appropriate payment file (timestamp instead of file reference inside ACK file)

Email subject:

The payment file HSBC received at <Timestamp> was in error

(<> indicates dynamic fields)

Example: The payment file HSBC received at 2016-09-16-14.54.09.674631 was in error

Email body:

The payment file HSBC received at <Timestamp> was in error due to the following reason: <CNTL error message>.

Example: The payment file HSBC received at 2016-09-16-14.54.09.674631 was in error due to the following reason: File header missing.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | HSBC |  |  |  |  |
|  | L |  |  |  |  |
|  |  | Archive |  |  | Archive |
|  |  | Errors |  |  | Files in error and error log |
|  |  | Files |  |  |  |
|  |  | L | FromHSBC |  |  |
|  |  |  | L | Acknowledgements | Input folder |

## Reporting – ISACOM100

Ascentric is obliged to submit an annual return containing information detailing all the ISAs managed within the period to HMRC.

For some time Blue Button and Sonata systems work on production simultaneously – this is referred to as dual running.

No transaction history is being migrated and clients are migrated on a phased basis throughout the tax year, thus the return needs to be produced across Blue Button and Sonata.

Blue Button return is the HMRC defined spreadsheet/schema (.xlsx file) whereas the Sonata return is a standardised report output based on HMRC flat text file specification ( .txt file). However, the return submitted to HMRC needs to be in the same file type, same format and any duplicates need to be removed/consolidated (this is a regulatory requirement). Therefore, BOT performs transformation of both returns to get them into a similar format so that the manual merging is much simpler.

ISACOM100 file processor waits for an input Excel file. Once the file has been detected, the processor converts Blue Button client IDs into primary Sonata client IDs using the RIMA and ODS databases. If conversion is not possible, then Blue Button client IDs are left.

The same ISACOM100 file processor also waits for an input .txt file. Once the file has been detected, the processor converts the file into Excel format based on HMRC schema. This consists in, among others, mapping Sonata records of type 2 to the corresponding columns in the HMRC schema and adding column with primary client ID using account number and ODS database.

Both output files are then manually merged and deduplicated.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | BackOfficeTools |  |  |  |  |
|  | L | DualRegIsaCom100 |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromTechnical | Input folder |
|  |  |  |  | ToSonata | Output folder  Please note that the folder name is misleading – the output files are not consumed by Sonata |

## Reporting – ‘Other interest Return (Type 18)’

Ascentric is obliged to submit an annual return containing information detailing the interest a person has received on their investments/savings by client and by ISIN to HMRC.

For some time Blue Button and Sonata systems work on production simultaneously – this is referred to as dual running.

No transaction history is being migrated and clients are migrated on a phased basis throughout the tax year, thus the return needs to be produced across Blue Button and Sonata.

Blue Button return is the HMRC defined spreadsheet/schema (.xlsx file) whereas the Sonata return is a standardised report output based on HMRC flat text file specification ( .txt file). However, the return submitted to HMRC needs to be in the same file type, same format and any duplicates need to be removed/consolidated (this is a regulatory requirement). Therefore, BOT performs transformation of both returns to get them into a similar format so that the manual merging is much simpler.

Other Interest Return Type 18 file processor waits for an input Excel file. Once the file has been detected, the processor converts Blue Button client IDs into Sonata client IDs using the RIMA database.

The same Other Interest Return Type 18 file processor also waits for an input .txt file. Once the file has been detected, the processor converts the file into Excel format based on HMRC schema. This consists in, among others, mapping Sonata records of type 2 and 3 to the corresponding columns in the HMRC schema, ensuring the type 3 records (client detail) are combined with the corresponding type 2 record in the .xslx report and adding column with Sonata client ID using client’s title, forename, surname and post code and ODS database. If this lookup fails, then the column is left blank for manual lookup.

Both output files are then manually merged and deduplicated.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | BackOfficeTools |  |  |  |  |
|  | L | DualRegOIR |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromTechnical | Input folder |
|  |  |  |  | ToSonata | Output folder  Please note that the folder name is misleading – the output files are not consumed by Sonata |

## Reporting – ‘Product Sales Data’ report

Ascentric is obliged to submit a retail investment report to Financial Conduct Authority (FCA) on a quarterly basis. The report contains the following information:

1. new business (under code type 30, 21 or 17) which includes expected values of contributions and transfers,
2. drawdown cases (under code type 24) which includes single drawdown events,
3. transfers in (under code type 20) which includes values for all transfers in

For some time Blue Button and Sonata systems work on production simultaneously – this is referred to as dual running.

No transaction history is being migrated and clients are migrated on a phased basis throughout the tax year, thus the report needs to be produced across Blue Button and Sonata.

Blue Button report and the Sonata report are both .xml files based on FCA specification. However, only a single file may be submitted to FCA via online system GABRIEL and any duplicates (except for the ones for records of type 20) need to be removed (this is a regulatory requirement) in the following way: Blue Button record prevails and Sonata record is deleted and in addition any Bluebutton or Sonata records that do not have a corresponding match are added to the report.

Therefore, BOT performs transformation of both reports to get them into a single file.

PSD report file processor waits for two input .xml files – one from Bluebutton and one from Sonata. Once the files have been detected, the processor performs merging and deduplication.

The folder structure is as follows. The example is for TSH environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AcceleratorTSH |  |  |  |  |  |
| L | BackOfficeTools |  |  |  |  |
|  | L | DualRegPSD |  |  |  |
|  |  | L |  |  |  |
|  |  |  | Archive |  | Archive |
|  |  |  | Errors |  | Files in error and error log |
|  |  |  | Files |  |  |
|  |  |  | L |  |  |
|  |  |  |  | FromTechnical | Input folder |
|  |  |  |  | ToSonata | Output folder  Please note that the folder name is misleading – the output files are not consumed by Sonata |

## Reporting – RPSCOM100: Sonata TXT to XLSX and XLSX to Sonata TXT

Ascentric is obliged to submit an annual return containing information detailing all the SIPPs managed within the period to HMRC.

For some time Blue Button and Sonata systems work on production simultaneously – this is referred to as dual running.

Data for accounts not yet migrated to Sonata is available only in Blue Button, so to submit the complete annual return, it needs to be created based on both Blue Button and Sonata information.

The return created from Blue Button is an xlsx file, whereas the return created from Sonata is a flat text file ( .txt file). The final, complete return submitted to HMRC needs to be a flat text file (all details listed in respected documentation). As a result, there is a need to unify the types of the returns: the Blue Button one and the Sonata one (the Sonata flat text file needs to be converted to the xlsx file) in order for user to be able to merge the two sources of information, i.e. to remove or consolidate duplicates (this is a regulatory requirement). Then the merged, manually processed xlsx file needs to be converted again to flat text file, that is required by HMRC. Theoretically, the xlsx Blue Button file could be converted to flat text file in the beginning (one conversion instead of two), though the structure of the flat text file would not allow user to merge the two separate files as it is hardly readable for humans (columns are not separated and only defined by their length). Therefore, two BOT processors needed to be created: Sonata TXT to XLSX and XLSX to Sonata TXT.

Sonata TXT to XLSX file processor waits for an input txt file. Once the file is detected, the processor converts it from the txt to xlsx type, row by row. In case one of the rows is of incorrect length, such row is being skipped (it is not present in th output xlsx file) and the error details are added to the error log file. If the type of the input file or its structure (header or footer) is incorrect, the conversion is not feasible and both the input file and error log file are placed in the error folder.

The folder structure is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RPSCOM100** | | |  |  |
|  | L | **TXTtoXLSX** |  |  |
|  |  | L | **Files** |  |
|  |  |  | L | **Input** |
|  |  |  | L | **Output** |
|  |  | L | **Archive** | |
|  |  | L | **Errors** |  |

XLSX to Sonata TXT file processor waits for an input xlsx file. Once the file is detected, the processor converts it from the xlsx to txt type, row by row. In case one of the cells is incorrect, such row is not placed in the output file, and is placed instead in the error output file. Value of the incorrect cell is replaced by ‘XX…X’, where the number of ‘X’ letters is equal to the expected length of this field. The error details are also added to the error log file. If the type of the input file or one of the ‘header cells’ (the first columns of the first row) are incorrect, the conversion is not feasible and both the input file and error log file are placed in the error folder.

The folder structure is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RPSCOM100** | | |  |  |
|  | L | **XLSXtoTXT** |  |  |
|  |  | L | **Files** |  |
|  |  |  | L | **Input** |
|  |  |  | L | **Output** |
|  |  | L | **Archive** | |
|  |  | L | **Errors** |  |

## Model Portfolio account linking processor

Platform to Platform (P2P) data migration is a process responsible for transferring the definitions of models along with the links between models and accounts. To have the models and the model-account links successfully transferred, both model and model account link need to be successfully validated against a set of conditions.

Unfortunately, some models and model-account links have not been validated successfully and are not present in new version of Front Office application, making users of FO confused. Both models and model-account links need to be added by Customer Service, what generates lot of manual effort.

This BOT processor has been implemented to align missing or incorrect data efficiently. It waits for an input CSV file, which contains a list of account numbers, model template IDs and adviser outlet client IDs. Once the file is detected, the processor links the listed accounts to respective models unless a given account is already link to a model. The processor will validate if:

* Account, adviser and model exist in Platform,
* Particular outlet can attach account to model,
* Account type is matching to model’s asset set,
* Outlet has access to account,
* Account has DFM agreement in case of DFM model.

The folder structure is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ModelPortfolio** | | |  |  |
|  | L | **LinkAccount** |  |  |
|  |  | L | **Input** |  |
|  |  | L | **Output** |  |
|  |  | L | **Archive** | |
|  |  | L | **Errors** |  |