Scripts all documentation on github: <https://github.com/SamanGharatchorlo/Saman_Documents>  
And in dropbox: <https://www.dropbox.com/home/Service%20Delivery/Saman>

Script List

NNIP: xml2csv\_wrapper  
NNIP: xml2csv\_convert

Tanita: unavista\_mifid2\_wrapper  
Tanita: unavista\_mifid2\_xml2csv

Banco do brasil: banco\_xml2csv  
Banco do brasil: banco\_wrapper

Ruffer: RufferTransaction

Binck: SpaceRemoval  
Binck: RekeningID

NNIP automation process  
The NNIP Gaspode job has been automated through a python script. The gaspode xml config file ‘NNIP\_MIFID2\_Gaspode.xml’ runs the xml2csv\_wrapper.py script which takes a single xml file as an input (from the NNIP Incoming/PreprocessingMIFID prd folder) and runs the xml2csv\_convert script outputting a single csv file into the NNIP prd sftp Incoming/MIFID folder.

LGTVestra automation process

The gaspode logic could be encapsulated within the python script, therefore not requiring the stages 3&4 below. However if this is the case the python script unavista\_mifid2\_xml2csv.py would need to be adjusted to output the file with the correct name (see from lines 3472 in script).

Currently untested automation in uat

Note: The Fido jobs 1 & 3 are currently paused  
The LGTVestra input data needs to run through both a python script and then a gaspode job. All of the Fido/Gaspode/Ubu jobs, folders and xml files have been set but it hasn’t been fully tested and is stuck on stage two (details below). The full setup of this process goes as follows:

1. Fido: the ‘Client\_LGTVestra\_python\_MIFID\_prd’ Fido job pulls an xml file with a naming format TR.MiFIR.[0-9]{13}.xml e.g. ‘TR.MiFIR.20180111162943.xml’ from the LGTVESTRA sftp ‘Incoming/MiFID’ folder and drops it in the ‘D:\Data\ClientGaspode\LGTVestra\In’ folder in gas-prd.
2. Python: The xml file ‘D:\Data\ClientGaspode\LGTVestra\LGTVestra\_MIFID2\_python.xml’ runs the ‘unavista\_mifid2\_convert.py’ python script (using files with the naming format TR.MiFIR.[0-9]{13}.xml from the In folder i.e. from the Fido job above as the input) and renames the output file to ‘python\_processed\_*(yyyymmddhhmmss)*.csv’ and drops it in the LGTVESTRA sftp Incoming/MIFID.  
   *\*Currently the xml LGTVestra\_MIFID2\_python is in ModelFileLocations, but it is not being run and not coming up in the logs.*
3. Fido: the ‘Client\_LGTVestra\_gaspode\_MIFID2\_prd’ pulls the ‘python\_processed\_*datetime*.csv’ file from the LGTVESTRA sftp Incoming/MIFID and drops it in the ‘D:\Data\ClientGaspode\LGTVestra\In’ folder in gas-prd.
4. Gaspode: the ‘D:\Configurations\Gaspode\ModelFileLocations.xml’ file runs the gaspode configuration ‘D:\Data\ClientGaspode\LGTVestra\LGTVestra\_MIFID2\_gaspode.xml’ on the ‘python\_processed\_*datetime*.csv’ file in the In folder and drops the output in the LGTVESTRA sftp Incoming/MIFID renaming it ‘213800LGMAJK7USM1M98\_*(YYYYMMDD)\_(HHmmss).csv*’
5. Fido: the ‘Client\_LGTVestra\_MIFID2\_prd’ Fido job pulls the ‘213800LGMAJK7USM1M98\_*(YYYYMMDD)\_(HHmmss).csv*’ file from the LGTVESTRA sftp Incoming/MIFID and drops it in the ‘D:\Data\ClientIngestion\lgtvestra’ folder in ubu-prd.
6. Ubu: when a file is named ‘213800LGMAJK7USM1M98\_*(YYYYMMDD)\_(HHmmss).csv*’ the ‘D:\Data\ClientIngestion\LGTVestra\MIFID’ folder in ubu-prd the ubu job pushes the file into the clients GUI.

Binck Automation

Binck needs two python scripts to be automated, RekeningID and SpaceRemoval (processed in any order, although RekeningID then SpaceRemoval would be more efficient). These scripts could be combined into one to simplify the Fido jobs required.

Instructions from Anna Dziubanska:  
*Input files are being loaded to PROD SFTP for Binck: Incoming/preprocessingEMIR every day at 6.15 am.*

*They should be processed and uploaded to the same folder with slightly changed names – please add “EMIR\_” at the beginning of each file name.*

*The original file should also stay in the folder.*

These have not yet been automated.

Note: the python scripts used for NNIP Banco do brasil and LGTVestra are very similar. The main body and functions are the same (NNIP has a small addition where it uses the python library numpy to filter out certain trades, see function: filter\_ext\_trades in xml2csv\_convert.py line 2246). The main differences arise when it comes to naming the output files.