

IS Responses

- **Installation/Institution/ACH Originator Settings** – Based on a spreadsheet the client provides, we update the database settings for the client.
- **Interface Trading Asset Setups** – These are normally for mutual funds. We are provided a list of assets that we have to setup up in interface mutual fund trading asset setups.
- **Investment Model Account Connection** - We would run the script to connect investment models to accounts based off a spreadsheet provided from the client. The table name is investmentmodelsetting
- **Fee Regeneration** - Deleting fee information, such as generation batch, generation occurrence, and generated fees in that batch, to allow fee to regenerate
- **Registration/Location for Custody change** - Specifically on holdings and balancing transactions. Update registration and location based off information provided by client
- **Resetting passwords** - May not be appropriate for this list. Reset passwords back to the default used by IS for clients

Level 2 Responses

1. **Account Officer changes** (PTI1172 asks about once per month) – Using a spreadsheet from client, create a script that:
 - a. Updates active AccountRelationships for Administrator from IdentityRecordID X to IdentityRecordID Y.
 - b. Updates open Tasks for AssignedToUserGuid from UserGuid X to UserGuid Y.
 - c. Updates active Scheduled Tasks for AssignedToUserGuid from UserGuid X to UserGuid Y.
2. **Registration/Location Code** changes (KTC1129 asks once or twice per month; other clients ask randomly as they need it done en masse) –
 - a. Using instructions from client, change all holdings from Reg/Loc A/B to C/D.
 - b. Using spreadsheet from client, change certain holdings from Reg/Loc A/B to C/D.
 - c. Both options are currently done by changing the Lot and all Transaction/TransactionHistory records. So it does not retain history of Reg/Loc for As Of reporting. It would be more accurate if there was a method to create the Reg/Loc Change transactions, which is too much work to do by script.
3. **Unreconcile or unvoid checks.**
4. **Statement Recipient setup** (KTC1129 asks every other month or so; HIM1190 has asked a few times) – Using a spreadsheet from client, create a script that adds a Statement Setup to the provided accounts for recipient X.
5. **Close accounts** – Using spreadsheet of accounts from CBM1061 on a monthly basis:
 - a. Backdate the Trade, Settle and StatementEffective Dates of the sweep transactions for the final sweep asset cash interest payment to the previous month end date.
 - b. Update the accounts with a closed date of the previous month end date, and set the status to Closed.
 - c. They almost always provide an account that should not be done. Validation that the account balance is 0 (or P+I nets to 0), and all transactions are <= closed date is a must!
6. **Institution/Installation Settings** – Random requests to change various settings.
7. **Assets on Approved List** – Using a spreadsheet from the client, write a script that sets ALL assets to Not Approved, then set the assets on the spreadsheet to Approved. Spreadsheet can be a

mix of Cusips and Symbols, some of which are not actually in the client database (they're getting the asset list from some outside source it seems).

8. **Update Matrix Status in the Interface Setup** area to Active so an asset can be traded.
9. **Change FeeSetting on Fee records**. Client reports they cannot close an account, or the outstanding fee balance is not collecting. This is usually due to the client removing the account from the fee group before clearing the fee balance. Afterwards they can't manually pay against that Fee Setting, and they can't get back to that same FeeSettingId. Occasionally this is due to the fee being in Income but they have allocated a payment to Principal.
10. **Pool Factor Inserts** – this would be for **missing** pool factors
 - a. Locate the pool factor on Refinitiv's site
 - b. Insert data to PoolFactor table (AssetId, Factor, PaymentDelay, DateEffectiveFrom)
 - c. Update previous pool Factor record's DateEffectiveTo date

Monthly

- **Indices** - run script to identify missing/wrong/Dates/Values, lookup prices, write update script, get approved, send to IT
- **IAD's** - run script to identify missing/wrong/Values, add/correct values, write update script, get approved, send to IT
- **Pricing Updates**

Weekly

- **All Caps for Issuer** - write script, get approved, send to IT
- **Pricing Updates**

Daily

- **Pricing Updates**

Accunet – These would require API connections to Accunet, unless they are the result of bad data in Cheetah.

- **500 errors** - research, write a script to update accunet-db_identity database
- **Duplicate Assets from gold to Cheetah** - identify duplicates, write script, get approved, send to IT
- **Duplicate HoldingMasters** - identify duplicates, write script, get approved, send to IT
- **Role Updates** - Identify duplicate or mismatch of roles and remove them by script
- **Duplicate Account Master IDs** - identify duplicates, write script, get approved, send to IT

********* - A process that can be done via current API library.

********* - A process that is viable via API, but does not yet exist in the current API library.

********* - A process that requires multiple steps or should have all or part protected from general user manipulation (i.e., ASC only). Best as enhancement as a PUT call. If protected, not found on Swagger page, and only callable by ASC service accounts.

********* - A process that requires steps not allowed by the application in "normal mode". This is likely not going to change. This would be something PM may want to explore to better understand the true use case to create an enhancement if it is something that should be allowed.