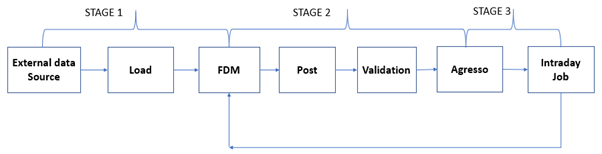
**Basic concepts of FDM**

This Document demonstrates the basic structure of the FDM system.

**What is FDM:**

FDM stands for “**Finance DataMart**” system which is generally a multidimensional database (called as Cube). The FDM will enable Finance change by providing Finance with a store of data; an application for system generated calculations; the ability to post data to, and extract data from, Agresso; and an analysis and reporting functionality. The purpose of FDM is to import data into Agresso.

The below diagram illustrates the stages of data flow throughout the FDM system:



Description of each stage is as follows:

**STAGE 1:**

This stage is about loading the data from an outside source to FDM system.

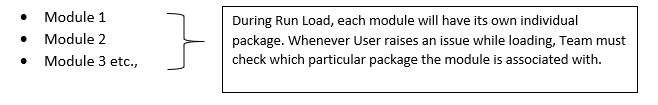
**External data source:** An outside source can be any excel sheet, data from a database etc.

**Load:** Moving the data from source to FDM.

**FDM:**

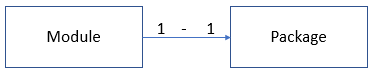
FDM system contains processes which would act as an interface for moving the data from source to FDM.

PROCESSES consist of modules, Allocation Engine and Cube Calculations:



Each Module has its own individual package that follows a **one-to-one** relationship.

* Name of the packages are same as the names of modules which makes it easy to identify a package during analysis.



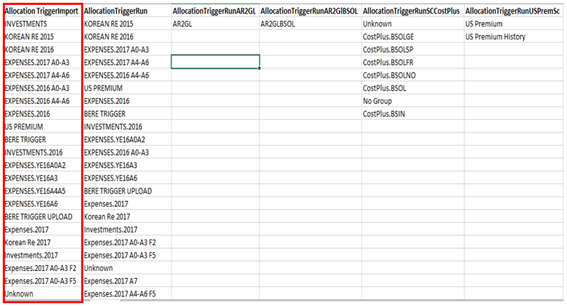
One -to- One Relationship

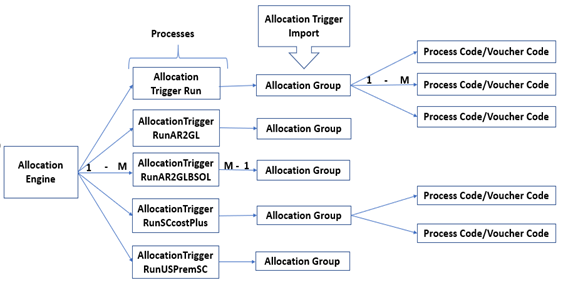
* **Allocation Engine**: The allocation engine enables a user configured set of rules (Allocation Rules) to be applied to any process that requires data to be allocated.
* Allocation engine provides the Finance Team with the ability to easily update allocation rules to reflect business changes.
* The allocation engine is flexible, scalable, and fully responsive to the demands of both real-time actual data and scenario modelling for the budget and forecast process.
* This Module has few Processes. During a Load, if we are unable to find a module in the first set of generic modules, then that can be found in Allocation Engine (Load).

Allocation engine has a set of allocation groups that can be selected while loading for a selected period which are listed under Allocation Trigger Import as highlighted in the below image.

Below are the allocation processes which uses the allocation groups from Allocation Trigger Import as selected by the User.

1. AllocationTriggerRun
2. AllocationTriggerRunAR2GL
3. AllocationTriggerRunAR2GLBSOL
4. AllocationTriggerRunSCcostPlus
5. AllocationTriggerRunUSPremSC





* **Cube Calculations:** All the above modules will be an input for this Cube during Posting/Analysis as all these are inter-related. All the data that has been loaded into the above modules will be calculated in the Cube and available for Posting/Analysis.
* The Finance processes generally depends on a series of ad hoc databases and spreadsheets to prepare data for posting into General-Ledger system, resulting in process delays and several control issues.
* FDM has a calculation engine to replace the ad hoc spreadsheets and databases used to prepare data for posting to the General-Ledger system. This reduces the time taken to complete the period end process and enhance control.
* In addition to handling the import of data and calculation rules from multiple sources, the engine performs real-time and scheduled calculations including earnings, IBNR, claims handling expense, and FX revaluations.

**General Ledger:**

A general ledger represents the record-keeping system for a company's financial data with debit and credit account records validated by a trial balance.

**STAGE 2:**

This stage is about posting/delivering the data from the FDM cube, post the data into Agresso by performing Validations.

**Post:**

Once the data has been processed in the FDM, it needs to be pushed into Agresso so the end users can access the data from Agresso. Process of committing the data from FDM to Agresso and pushing it back to FDM is called “Posting”.

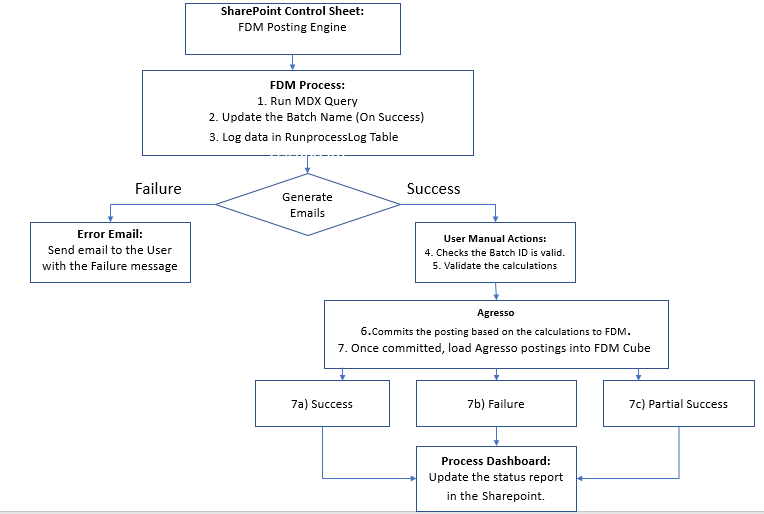
**Posting engine** will serve the purpose of pushing the data from FDM to Agresso.

**Posting Engine**: Once the data has been calculated in the Cube, user performs Posting from FDM Cube and commits the Posting to Agresso, by verifying data with “All calculations” and if the values are correct. At the backend, Posting Engine will run appropriate posting queries based on the processes defined in the engine. All this processing occurs on the FDM Server and will automatically create all outputs that will be committed into Agresso.

**Note:**

A SharePoint is used to hold these posting queries so that if the MDX for a Process needs to be updated or a new Process needs to be configured, this can be done through SharePoint without the need to redeploy SQL Server Code.

The Posting Engine process is outlined as below diagram:



**Validation:**

Validation needs to take place before the data is committed to Agresso.

Here Data is validated in the excel pivot on comparing with the data generated from base allocation, Live Agresso and true up.

**Agresso:**

It is a database where data is imported from FDM cube and is used by end users. This can also act as an input to FDM sometimes.

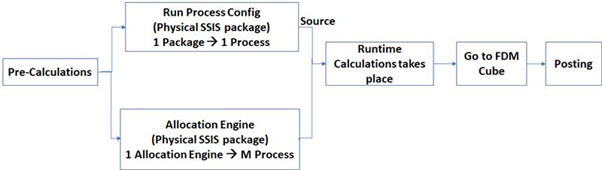
**STAGE 3:**

The Committed data from Agresso will be imported back into FDM through a Job called “Intraday”

**Back in FDM:**

All historical and current postings in Agresso are populated into the FDM Database and visible to end users through the FDM Cube using excel pivots. The data is updated on a periodic basis intra-day job(generally every 10 minutes) and all new transactions, if committed from Agresso , will be reflected in FDM.

**What are Pre-Calculations and Run-time calculations:**



**Pre-Calculations:**

These are sub-divided into 2 types:

1. **Run Process config:**

Here each process is allocated with its own individual physical package that would get executed during Load.

1. **Allocation Engine:**

Here each module is allocated with its own individual physical package which runs for multiple process.

There are few packages(Ex: US premium) which are 1-1 but use the Allocation engine internally.

**Run Time Calculations:**

Pre-requisite for any Run time Calculations is Pre-calculations have happened and are fed to Run time Calculations.

**Note:**

Not all the Modules will have to go through the Pre-calculations. Some of them can directly processed in Run time calculations for which Pre-calculations acts as a source.

Ex: “RI earnings” doesn’t have pre-calculations and it gets processed directly in the Cube(Run time calculations) but to get RI earnings calculated Cube needs data from “RI spend” which is part of pre-calculations.

**Possible Scenarios where the issue can arise during the Post or Load:**

1. Technical failure: There could be multiple reasons for a technical failure when user tries to Post or Load likely:

* Disk Space/Memory issue.
* An error in the System or Package.
* Server crash.

1. Load or Post did not work as expected: Here the load or post will be processed without any issues in the system, but the data will be a mismatch between the FDM and Agresso.

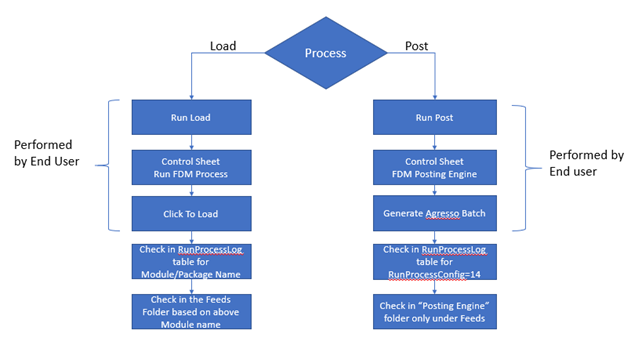
**Pre-Requisites for Load and Post:**

* System Checks
* Data checks
* Data has been posted

**Post Checks to validate Load and Post:**

* Data checks
* Load/ Post has successfully completed
* Posting has come back to FDM
* Mismatch in Process codes selected for Load and Post
* Data has been populated after the load.

**General Flow of Load and Post Process during Analysis:**

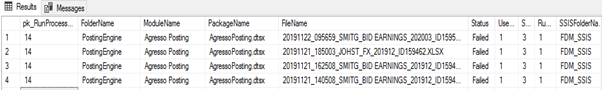


The selections of the process made by the User in the Run FDM Process Control sheet to load the data into FDM are called “**Parameters to Load**” and those selected in the FDM Posting engine Control sheet to get data from FDM into Agresso are considered to be the “**Parameters to Post**”.

**Analysis during issues in Load or Post**:

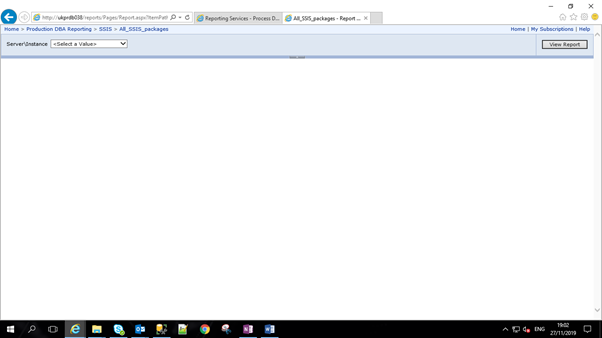
First check if the User has made a Run Load or Run Post by looking into the “fk\_RunProcessConfig” column of the Table called RunProcessLog (FDM\_Process DB). If the Column value is anything other than 14, consider that as a Load. If the Column value is equal to 14, it is a Posting by User.

1. **Load**: Details like the Folder Name & Package Name can be retrieved from the Tables namely “RunProcessLog” and “RunProcessConfig”. With this information, Team can check that relative package logs to investigate further on the issue.
2. **Post:** As said above, that during the Post, Fk\_RunprocessConfig value will be 14 due to which the Package and the Folder Name would be the same throughout any postings. Package “AgressoPosting” and FolderName “Posting Engine” are used to identify the root causes of any issues raised by User while Posting.

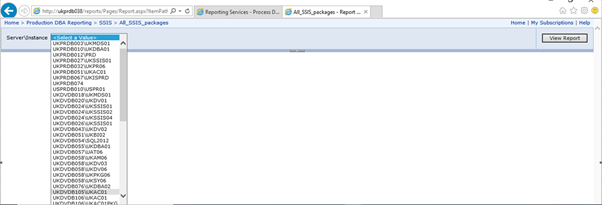
Check Table **“RunProcessLog”** for any Failure in the Load/postings:

* From the above table, details like Package Name for the respective run can be identified.
* Based on the Package name, Check the related logs in SSIS packages sharepoint as below:

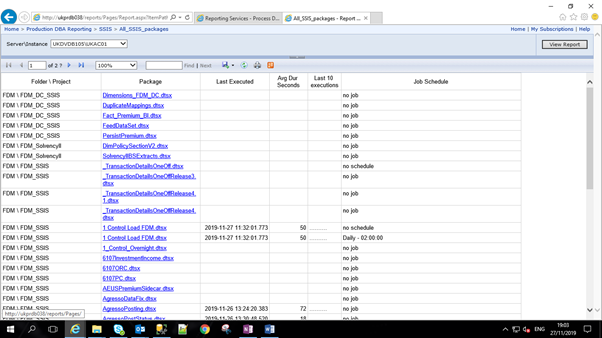
http://ukprdb038/reports/Pages/Report.aspx?ItemPath=%2fProduction+DBA+Reporting%2fSSIS%2fAll\_SSIS\_packages



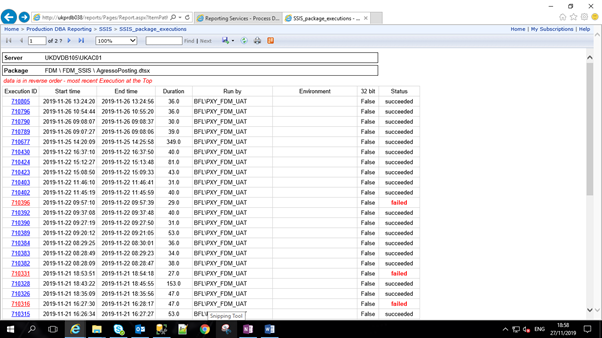
* In the Sharepoint, Select the particular server (PRD/UAT/SYS) and click on “View Report”.



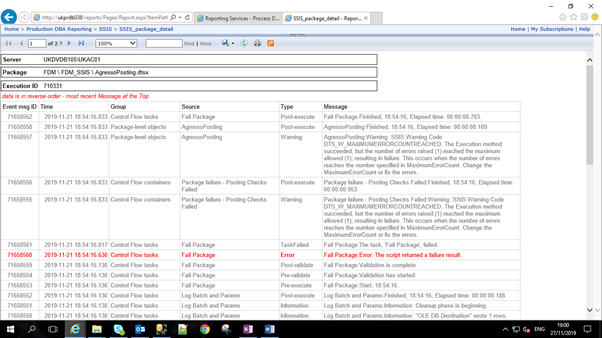
* Once the server has been selected, it will list out all the packages associated with that system.



* As Package Name will already be identified in the above process, click on that package.(Ex: In the Above RunProcessLog table, Log shows that the file related to posting has failed and its related package is “Agresso Posting”)
* This will list out all the executions of the package. Failed run will be highlighted with Red as below.



* Click on the Execution ID which will list out all the message details of the failure including the stage at which failure occurred in the package.



**User Access Request Process:**

New users should be given FDM access when their user account is set up

(if requested in the new joiner form completed by their manager).

They are then added to the AD group and don’t need to request access separately.

If this doesn’t happen their line manager should request access.

**2a)** If they write directly to Service desk, SD create a ticket that then routes the request to Finance management for their approval before the

SD analyst adds the user to the AD group (this is the actual approval step).

**2b)** If they write directly to FDM support, Team to send an email to SD to raise new ticket for providing the access by taking approval from respective manager beforehand.

SD then send notification – sometimes to the requestor and sometimes to the person who gained access and sometimes to both.

Whenever FDM Support receives a request from a user asking to provide access for Sharepoint/Feeds folder, Team to raise request with SD to provide the access to the User by adding them to AD group as below

Whenever FDM Support receives a request from a user asking to provide access for ControlSheets/Posting engine, Team will ask User to provide a mirror account for providing permissions on control sheet and according grant the same.

**AD groups:**

**View FDM -** BFL\App.FinanceDataMartCube.RO

**View TDM -** BFL\App.TriangleDataMartCube.RO

**Feeds folder -** \\bfl.local\uk\Groups\Finance\Reporting\FDM\Feeds

**ViewSharepoint/ViewControlsheets-** BFL\sec.w2kfileserver.groups.finance.reporting.fdm.feeds-rw

**Access Control sheets(process which are inside control sheets) –** Permissions given byFDM team

**Use the posting engine -** Permissions given byFDM team

**Frequently Asked Questions to Client:**

For which Process code did the User has committed the Posting?

Has the User loaded the data into the FDM before committing the Posting?

Has the User verified the values after loading/posting data ?

Has the User posted the data for all process codes(if more than one Process code selected) that was chosen by User during the Load?

For which Process code/ Account code/ Entity code did the user has run Load/Post ?