

1. Load tonnage per pod. Adam hasn't added unit pod information to database and API yet. Based on current understanding of business process, dispatch is responsible to split a load sheet to each pod upon the factors include road ban, pod capacity, legal limit, etc. We need add pod split function in the product haul scheduling function on Rig Board.

TO BE CLARIFY:

- In program, same blend may be used in different stage/plugin/interval, they are showing on the Rig Board separately. Currently dispatch has to schedule them one by one, each blend will have its own load sheet. Bulk plant will blend them sheet by sheet, Is it right? combining multiple load sheet and blend together is not allowed.
 - Once we have blend tonnage in load sheet, do we still necessary have to split one load to multiple cuts?
 - Answer from Colin: We have installed Blend Train in all bulk plant, it is enough to blend once for a single bulker or a B-Train.
 - Do we know third party bulker pod information? How many pods in a bulker? What are the sizes?
 - B-train has LEAD and PUB units, each of them has own serial number. LEAD may have up to 3 pods, and PUB always has 2 pods. Are LEAD and PUB always used together? If yes, there will be up to 5 pods. So when dispatch schedules one product haul using a B-Train, he has to calculate tonnage for each pod. Is it right?
 - Answer from Colin: It is possible only use LEAD. All LEADs have 2 pods. Ideally dispatch calculate the load tonnage for every pod.
2. Load information post back. Once a load sheet is started to blend, a BLENDING status status should be able to push back to backend, so other location will see the status to avoid the double blend. Once the blending is done, and the COMPLETE status must be pushed back to backend, so it won't listed to any location todo list.
 3. Blended load sheet can be pushed back to backend with updated details, backend may compile movement journal and post to AX. Movement journal is per blend load, it is strictly aligned with a load sheet. When backend received the completed load sheet, it will compile a movement journal export file. AX will pick up the export file to post the journal. If the post fails, the export file will be moved to error folder. A new business process is needed to monitor the post failures and repost.
 4. MTS. When we look at detail information in MTS, it is as same as updated load sheet. We need the pod allocation information anyway. So it depends topic 1. There will be development dependency and cannot implement in short term. Suggested solution: 1. Keep current business process, copy load sheet information from screen to printed MTS book. 2. Neville will implement a simple function to print what shows on screen.

TO BE CLARIFIED.

- Since each load sheet is for one blend on one unit, so if we have a product haul carry three blends, that means we will have three load sheets for the unit. Should we combine them to one MTS? Or we can use three MTSS. Further, if three blends will be hauled to three locations, it seems we have to have three MTSS. This is a business rule to be clarified
5. Offline data caching facility. To support weak WIFI coverage scenario, We need local caching for master data, inventory checking, load sheet and also message queue for post back loaded information.

6. Product name consistency. Assure load sheet is using AX product name which associate to an IIN number.

To satisfy above requirement, here are the solutions in applications.

1. Enhance eServiceOnline Product Haul function to add pod allocation.
 - a. Add unit pod information master data.
 - b. User interface improvement to split tonnage to pods
 - c. Business rule implementation to validate load amount against pod capacity, legal limit and road ban factors.
 - d. Enhance load sheet structure to support pods
 - e. Enhance load sheet printing to support pods
 - f. Enhance load sheet API expose data to Blend Verification App
2. Add Inventory Checking API to query AX by location and product name.
 - a. Fix GetLoadSheet API to assure product names is using AX product name.
 - b. Create inventory view in SanjelData database to connect to AX database.
 - c. Build Inventory Checking API
3. Add Load Sheet status update APIs.
 - a. Add BLENDING, COMPLETE status to load sheet
 - b. Add BLENDING status update API
 - c. Add COMPLETE status and load sheet details post back API
 - d. Improve Rig Board to show BLENDING, COMPLETE status
 - e. Filter out COMPLETE load sheet in get load sheet API
4. Integration for posting movement journal to AX
 - a. Add integration to compile movement journal export file.
 - b. Add import movement journal function in AX.
 - c. Add new function to handle post failures and repost.
5. Local cache facility
 - a. Update inventory data, master data, load sheet cache from server in a frequent cycle.
 - b. Expose master data through local API
 - c. Query product inventory through local API
 - d. Query and update load sheet through local API.
6. MTS
 - a. Port load sheet printing code to print MTS on local.
 - b. Loaded load sheet database implement as MTS central repository.

No. 2 might be able to implement with small amount of effort. For other items we have to look into details and make estimate. The priority will be determined by management team.