**Driver Login calls DriverLoginProcess, PreferencesProcess, CodeTableProcess, CommodityMasterProcess, ContainerChangeProcess, TerminalChangeProcess, TripInfoProcess**

**Service: DriverLoginProcess**

Status: Fully functional

Required Fields: EmployeeId,Password,PowerId,Odometer,LoginDateTime, PndVer

Special Fields: OverrideFlag = Y if driver has double checked Odometer and is resubmitting

?? Fields: LocaleCode, MdtId

Fields returned: TermId, RegionId, AreaId, TripNumber, TripSegNumber, DriverStatus

Note: New System Preference DEFRouterPath must be a valid path.

Four packets of data will be created in or appended to a GPSScrapPkt.x file in the Router\Send folder and then an attempt will be made to rename the file to GPSScrapPkt. Currently only one attempt will be made.

Possibly, another service should attempt to rename the file.

What should be done if the file cannot be created?

**Service: PreferencesProcess**

Status: Fully functional

**Service: CodeTableProcess**

Status: Fully functional

**Service: CommodityMasterProcess**

Status: Fully functional

**Service: ContainerChangeProcess**

Status: Fully functional

Special Fields: LastContainerMasterUpdate, populated to receive latest changes

LastContainerMasterUpdate , not populated to receive all containers.

**Service: TerminalChangeProcess**

Status: Fully functional

Special Fields: LastTerminalMasterUpdate, populated to receive latest changes.

LastTerminalMasterUpdate , not populated to receive all terminals.

**Service: TripInfoProcess**

Status: Functional

Open Issues: Need to be able to determine if we are sending trips because of a login request or because a new trip has been entered or an existing one modified for a driver that is logged in. Currently set to process login request.

**Driver Trip Acknowledgement calls DriverTripAck for each trip.**

**Service: DriverTripAckProcess**

Status: Fully functional

**Load/DropContainer action calls DriverContainerActionProcess**

**Service: DriverContainerActionProcess**

Status: Functional

Special Fields: ActionType: L=Load, S=Dropped

MethodOfEntry: M=Manual, S=Scanned.

**Enroute action calls DriverEnrouteProcess**

**Service: DriverEnrouteProcess**

Status: Functional

Limitations: Code has not yet been written to resequence trips for driver if driver has gone out-of-sequence.

Other Issues: Need to determine how to do deletes and updates and inserts before anything is committed.

**Arrive action calls DriverArriveProcess**

**Service: DriverArriveProcess**

Status: Fully functional

**Segment Done (Stop Transaction, Yard/Scale, Scale) calls DriverContainerActionProcess for each container and DriverSegmentDoneProcess for the segment completed.**

**Service: DriverContainerActionProcess**

Status: Functional

Limitations: Code to process the “quantity in container number” has not been written yet.

Testing needs to be completed for setting of the location warning flag.

Testing needs to be completed for exception and review processing.

Special Fields: ActionType: D=Done, E=Exception, R=Review

ActionCode: For Yard, Review Reason Code or Exception Code

ActionDesc: For Yard, Review Reason Description or Exception Description

SetInYardFlag: For Yard, Y/N

**Service: DriverSegmentDoneProcess**

Status: Functional

Limitations: Testing needs to be completed for exception and review processing.

Note: Exception on a PF followed by a RT segment, will result in the question,

“You could not process some containers. Do you still need to return to a yard?”

If yes, call DriverSegmentDoneProcess with ActionType of D=Done.

If no, call DriverSegmentDoneProcess with ActionType of X=Canceled.

**This service includes the MarkTripDone processing.**

Status: Functional

Limitations: Testing needs to be completed for exception and review processing.

Code has not yet been written to check for exception delay type. If any delay on the trip was an exception type, change the status of the segment to E=Exception.

Code has not yet been written to check if trip should be placed in the error queue. This is a preference.

Code has not yet been written to re-sequence trips or the driver, now that one has been completed.

**Driver Add a Return-To-Yard Segment**

**Service: DriverSegmentDoneProcess**

Status: Functional

Limitations: Testing add RT/RN segment is in progress.

Note: ActionType: P=Pending, DestCustHostCode, TripSegType: RT or RN DriverGenerated:Y

**Service: DriverContainerActionProcess**

Status: Functional

Note: Action Type: A

Container information comes from the same containers loaded or picked up on the previous segment. If there are none, RN is specified in the DriverSegmentDoneProcess and DriverContainerActionProcess is not called.

**Driver Changing a Return-To-Yard Segment**

**Service: DriverSegmentDoneProcess**

Status: Functional

Limitations: Testing change RT segment is in progress.

Note: ActionType: P=Pending, DestCustHostCode, DriverModified:Y

**Driver Delay and Back on Duty Actions calls DriverDelayProcess**

**Service: DriverDelayProcess**

Special Fields: ActionType = D=Delay, B=Back On Duty

Status: Functional

Other issues: What do we do if there is no driver status record because driver is not logged in.

This should never happen. But should we create one and add a driver history record?

On a back on duty, if there is no delay record started, should we start one and end it?

**Driver Fuel Entry calls: DriverFuelEntryProcess**

**Service: DriverFuelEntryProcess**

Status: Fully functional

Open Issues: None

**Driver State Line Crossing calls: DriverStateLineProcess**

**Service: DriverStateLineProcess**

Status: Fully functional

Open Issues: None

**Driver New Container: calls DriverNewContainerProcess**

**Service: DriverNewContainerProcess**

Status: Fully functional

Special Field: ContainerNumber, always populate even if it was not changed.

**Driver Sends a Message: calls DriverMessageProcess**

**Service: DriverMessageProcess**

Status: Fully functional

Fields: SendId field and other fields are populated.

Open Issues: None

**Driver Request Messages: calls DriverMessageProcess**

**Service: DriverMessageProcess**

Status: Fully functional

Fields: SendId field only field populated.

Open Issues: None

**Driver Send Power Unit Change or Odometer Correction**

**Service: DriverOdomUpdateProcess**

Status: Fully functional

Fields: Send unchanged power id to use odometer update processing. ”Correct Odometer”

Send new power id to use power unit changed processing.\_”Change Power Unit”

**Driver Send Picture or Signature**

**Service: DriverImageProcess**

Status: Functional

Note: System Preference DEFSignatureCapturePath must be a valid path.

Fields: ImageType: S=Signature, P=Picture

**Driver Send GPS Location**

**Service: DriverGPSLocationProcess**

Status: Functional

Note: New System Preference DEFRouterPath must be a valid path.

After adding an entry to the new GPSLocation table, a GPSScrapPkt.x file for TRACKER will be created in the Router\Send folder and then an attempt will be made to rename the file to GPSScrapPkt. Currently only one attempt will be made.

If a file cannot be written, the GPSendFlag will be set to 3 (SendFailed).

Possibly, another service should attempt to resend the packets with a GPSSendFlag of 3 and also attempt to rename the file.

**Driver Logoff**

**Service: DriverLogoffProcess**

Status: Functional

Fields: If driver has more trips, populate the trip and segment fields with the next trip.

Note: TripNumber, TripSegNumber, DriverStatus do not need to be populated.