Generate a trial GL Trial Balance xml integration file (inbound to M3)

# Outline

At this point we are looking to test the MEC component of the end solution process. Therefore we are generating an xml file from the source system (AS400) manually. We know that the connection and extraction of the source data will be done by Informatica at the end solution but for this trial we plan to manually piece together a hypothetical import file for MEC. The goal is to test that MEC can pull in a file and that it can generate the GL trial balances within M3 correctly.

## XML structure

The typical way MEC receives xml files is with an Envelope which contains a header and a detail (or body). The Envelope is simply a wrapper to contain the entire message. Theoretically any 1 xml file could potentially contain multiple messages and each would be wrapped by an envelope. The envelope would be the device used to separate one message from another.

Example:

<Envelope>

<Header>

<HeaderData>

</HeaderData>

</Header>

<Detail>

<DetailData>

</DetailData>

</Detail>

</Envelope>

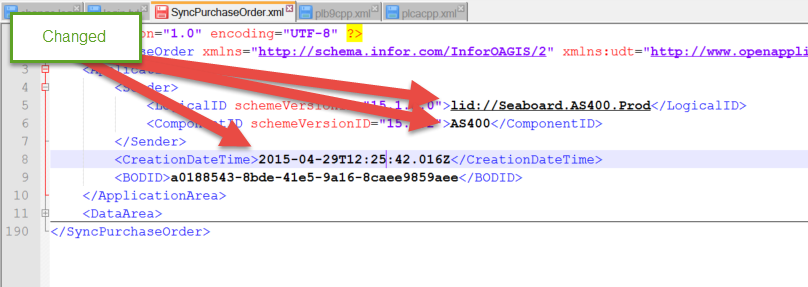
### Envelope

For the envelope we took a standard M3 outbound message and ‘stole’ it’s envelope. This was convenient because the MEC PAT tool has definitions already defined for standard M3 outbound document envelopes. We changed the original from <SyncPurchaseOrder> to <SyncGLTrialBalance>



### Header

In the original M3 header there is information for <Sender>. This is originally setup as “M3BE”. We adjusted this to be the AS400



### Detail (Body)

In the original M3 detail it identifies that the message is a purchase order. We needed to adjust this to GLTrialBalance. After this GLTrialBalance segment we changed the PurchaseOrderHeader to GLTrialBalanceHeader and the PurchaseOrderLine to GLTrialBalanceDetail. Inside GLTrialBalanceHeader segments we inserted the xml data from ‘plb9cpp.xml’ file. Inside the GLTrialBalanceDetail segment we inserted the data from ‘plcacpp.xml’ file.





