**Service Billing Back Feeds**

**INDEX PAGE**

1. **Invoice Status Backfeed – Updates Thirsty with status AP Acceptance**
2. **Thirsty Update – Backfeed File – Creates ~~Invoice~~ object in Thirsty  
   Creates a \*CUSTOMER\* Invoice object in Thirsty (Name: Customer\_Billing\_\_c). \*Invoice\* objects are Service Provider invoices which get created when a service provider bills Coca-Cola for their services. CM and BRIM are not involved in creating those, it’s a core Thirsty function.  
   The Invoice object does get queried by CM, first to generate Customer Billing BITs for CC, then below just because the invoice id is needed for the Customer Billing Invoice.  
   It’s important not to get these two mixed up since they are very different things. -- Stefan**
3. **AP Payment Backfeed – Updates Thirsty with status ‘Invoice Paid’**
4. **Free Call Backfeed – Updates Free call to Thirsty**
5. **Workflow to restart Thirsty rt**

* **InvoiceStatus BackFeed (updates Thirsty with AP Acceptance status)**

**Paths :**

|  |  |
| --- | --- |
| (PAYSP puts files to below Location) |  |
| UPDATE\_THIRSTY\_COLLECT\_DIR | /usr/sap/interfaces/MZP/ServiceBilling/UpdateThirsty |
| UPDATE\_THIRSTY\_COLLECT\_FILENAME | Workorders.\* |
| Invoicestatus.splitfiles WF collects from above and |  |
| (archives here – without processing)  INVOCIESTATUS\_ARCHIVE | /usr/sap/interfaces/MZP/ServiceBilling/UpdateThirsty/archive |
| AND ALSO |  |
| Splits in chunk of 100 and sends to  INVOICESTATUS\_BULK\_API\_PROCESS\_DIR | /usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/process |
|  |  |
| UPDATE\_THIRSTY\_REPROCESSING\_DIR | /usr/sap/interfaces/MZP/ServiceBilling/UpdateThirsty/~~Reprocessing~~  /usr/sap/interfaces/MZP/ServiceBilling/UpdateThirsty/  ( RECENT CHANGE ON 19TH APRIL 2014) |
|  |  |

**Workflows :**

* TCCC\_ServiceBilling.WFL\_InvoiceStatus\_SplitFiles
  + Input File to this WF is from SB >> PaySP flow.
* 3 TCCC\_ServiceBilling.WFL\_InvoiceStatus\_BulkAPI\_Batch
  + Billing (Invoice Backfeed)
    - TCCC\_ServiceBilling.WFL\_SplitInvoiceBackfeed.workflow\_1
  + Billing (Invoice Backfeed) - Reprocess
    - TCCC\_ServiceBilling.WFL\_ThirstlyUpdate.ReproBilling
  + **AP Payment BackFeed (changes status in Thirsty as ‘Invoice Paid’)**
    - SAP Program (ZBAR\_PAYMENT\_PROCESSING) writes files to

COLLECT\_S4\_SOURCE\_DIR /usr/sap/interfaces/SP0/outbound/AR/thirsty/data once a day

* + - 1 WFL\_CollectS4.Payment

Gets file from above and (Recently added) and

stores Archives to

/usr/sap/interfaces/SP0/outbound/AR/thirsty/archive (COLLECT\_S4\_MOVETO\_DIR)

And sends to

PROCESS\_INPUT\_PAYMENT\_DIR

/usr/sap/interfaces/MZP/ServiceBilling/backfeed

* + - 2 TCCC\_ServiceBilling.WFL\_Payment\_BackFeed\_Collect.Payment

Collects and splits and writes to

THIRSTY\_WRITE\_PAYMENT\_DIR /usr/sap/interfaces/MZP/ServiceBilling/backfeed/write

* + - 3.1 TCCC\_ServiceBilling.WFL\_ Payment\_BackFeed\_BulkAPI.payment\_backfeed

Collects from above and send to

* + - **REPROCESS\_INPUT\_PAYMENT\_DIR** (for the files having error response) **/usr/sap/interfaces/MZP/ServiceBilling/backfeed/reprocessing**
    - 3.2 TCCC\_ServiceBilling.WFL\_ Payment\_BackFeed\_BulkAPI.reproces
      * **Collect from REPROCESS\_INPUT\_PAYMENT\_DIR /usr/sap/interfaces/MZP/ServiceBilling/backfeed/reprocessing**
      * **And if fails goes back to reprocessing and error**

**Customer Invoice Backfeed Program**

**Updates Single Object (not BULK API)**

**– Create Invoice document linked with WO in Thirsty**

1. **WFL\_Collect\_S4**

Collects from

COLLECT\_S4\_SOURCE\_DIR usr/sap/interfaces/SP0/outbound/AR/thirsty/data

Sends to

PROCESS\_SPLIT\_BILLING\_DIR /usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty

And Archives at

COLLECT\_S4\_MOVETO\_DIR /usr/sap/interfaces/SP0/outbound/AR/thirsty/archive

1. **WFL\_SplitInvoiceBackfeed**

Collects from

PROCESS\_SPLIT\_BILLING\_DIR /usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty

And sends to

PROCESS\_SPLIT\_BILLING\_DIR /usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty/input

(done via code not via property file – just be aware )

1. **WFL\_ThirstyUpdate**
   * **Agent - Billing\_Input (Disk)**

**Collects the file from this location**

* + - PROCESS\_INPUT\_BILLING\_DIR /usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty/input
    - Filename : INV\_Thirsty\_output\*
    - **No Archive here – I think we should add   
      Files get archived when they are collected from CI. The only difference is that those have not been through the Split workflow. Another archive location here does not seem necessary to me. -- Stefan**
  + **Agent - CheckFileType (Analysis Agent)**
    - Navigates to Billing Path
    - Payment Path - Not used Mostly – Ask Ray  
      NOT used AT ALL -- Stefan
  + **Agent - Correlate (Aggregation)**

Creates 1 session for all lines of 1 invoice

The expected input for an invoice is a header record followed by separate billable items for labor, travel, parts and misc.

If parts more than 20 – sent in diff request maybe – Ask exactly what is that we r doing  
If there are more than 20 parts, the invoice is “split”. This means making a copy of the invoice record (RecType = H.) For example, if an invoice with 50 parts is received, the output to Thirsty will look like:  
Request 1:  
Invoice 12345  
 Parts 1 – 20  
Request 2:  
Invoice 12345  
 Parts 21 – 40  
Request 3:  
Invoice 12345  
 Parts 41 – 50  
Sending the same invoice multiple times does cause a little overhead but it’s rare that it’s needed, and it doesn’t cause any issues because the same fields get updated with the same values, just multiple times. -- Stefan

RecType = H (header), P(parts)..

**Sends to RT WFL – WFL\_ThirstyUpdate\_RT**

WFB\_Handler (Analysis Agent) – just forwarding ahead

**Send\_To\_Thirsty**

string invoiceQuery = "SELECT Id, Work\_Order\_Number\_\_c, Customer\_Name\_\_c, CustomerACN\_\_c, Invoice\_Status\_\_c, InvoiceDate\_\_c, Invoice\_Total\_\_c, PartsAmount\_\_c, SP\_ACN\_\_c, CustomerCallDateTime\_\_c FROM Invoice\_\_c where OrderNumber\_\_c = 'WORKORDERID'";

string partsQuery = "SELECT Id, ProductCode FROM Product2 where ProductCode in ";

string customerIdQuery = "SELECT Id FROM Customer\_Billing\_\_c where Work\_Order\_Number\_\_c = 'WORKORDERID'";

string laborCreditQuery ="SELECT Id, Counter\_ID\_\_c, Business\_Partner\_\_c, Used\_Call\_Number\_\_c, Customer\_ACN\_\_c, Call\_Limit\_\_c, Project\_number\_\_c, Remaining\_Call\_number\_\_c, Account\_\_c, Equipment\_Types\_\_c, Free\_Labor\_Credit\_Indicator\_\_c, Service\_Types\_\_c, Trouble\_reported\_code\_\_c, Sub\_Equipment\_Type\_\_c from Labor\_Credit\_\_c WHERE Counter\_ID\_\_c = '";

**consume block**

IF (udr type = InvoiceWrapper)

Retrieve the parts and make a parts query call to THIRSTY if even 1 part.

If no parts , proceed to check if counterid not null or na, then make call for Labor credit

If no parts or counterid , query workorderid (invoice query)

ELSE (it’s a response)

If it’s a parts query response

decode json response and put it in map

If parts are present – do a get call for customerid query (customerIdQuery) for each part in map

//To update used and remaning free call logic, adding new flow

laborCreditQuery

if (no\_parts)

"Response from parts, not having parts (not requiring customer ID) and not having a counter ID present, going to query info for Invoice");

//Handle Response for LC QUERY

//Handle Response for INVOICE (wrap.reqType == "REQ\_INVOICE")

If unable to decode response , **DISCARD record**

If able to decode response, make post call to Thirsty

wrap.reqType = "POST\_INVOICE";

rcu.Request.ResourceURI = "/services/data/v52.0/composite/batch";

//Response for composite request, Invoice + parts

* + - REPROCESS\_INPUT\_BILLING\_DIR

/usr/sap/interfaces/MZP/ServiceBilling/InvoiceToThirsty/reprocessing

Filename – Billing\_Input.SourceFilename

**Incoming requests, requests sent to Thirsty, and the responses from Thirsty are logged to a database table, TXN\_DETAIL\_LOG with category = ‘TYUPD’ and key = Workorder ID. Records are currently kept in this table for 3 months and those can be a great help in troubleshooting. -- Stefan**

* + Email : PROCESS FAILED for Invoice Number

**Free call Backfeed :**

**Cust1 :**

Workflows :

* + - Some SAP program generates files to /usr/sap/interfaces/SP0/outbound/AR/thirsty (cross check once this path )
    - 1 TCCC\_ServiceBilling.WFL\_CollectS4.FreeCall

Archives to

/usr/sap/interfaces/SP0/outbound/AR/thirsty/archive (COLLECT\_S4\_MOVETO\_DIR)

Then Moves to Output folder

/usr/sap/interfaces/MZP/ServiceBilling/backfeed

* + - 2 TCCC\_ServiceBilling.WFL\_FreeCall\_SplitFiles.workflow\_1

Splits and moves to

/usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/archieve

Send error email notification (does not write in error – Internal CM validation)

Free Call Loader Error Log - SP0

OutletACN is not present|1|1|88768904|FSC|4|N/A|N/A|BT|N/A|N/A||1120022567

* + - 3.1 TCCC\_ServiceBilling.WFL\_FreeCall\_BulkAPI\_Batch.wrkflow\_1
      * Writes to reprocess

FREECALL\_BULK\_API\_REPROC\_DIR String /usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/reprocess

* + - * Writes to error (error from Thirsty)

FREECALL\_BULK\_API\_ERROR\_DIR String /usr/sap/interfaces/MZP/ServiceBilling/BulkAPI/error

* + - 3.2 TCCC\_ServiceBilling.WFL\_FreeCall\_BulkAPI\_Batch.Reprocess
      * Same workflow as prev one

1. **WFL\_SBBackfeed\_script**

**This is for Recon. Sends email of the recon file to users.**

1. **WFL\_StartStop\_Script.** **sb\_backfeed\_RT**

**This is to stop and start the Thirsty related RT workflow as Thirsty session expires after an hour.**