Process Customs Manifest

# Step #1:

## Step #1.1:Set manifest from data base

Get Intl port code for manifest header

### [Go to Step #1.1.1](#Step_1_1_1)

[Step #1.1.1:](#_Go_to_)  get Intl code for header

(SELECT INTLCODE from TSAU\_CUSTOMS\_PORT WHERE CODE = ?)

**Query Parameters :**

1. CODE: from manifest header [from xml]()

Changes[1]: set INTLCODE in header CustomsRegPortCode [updated from db]

-Changes[2]:Set header channel “WEB”

Case#1: If Custom Port Discharge equal “0”

Get Intl port code for shipment header

### [Go to step #1.1.2](#Step_1_1_2)

[Step #1.1.2:](#_Go_to_step_7) get Intl port code for shipment header

(SELECT INTLCODE from TSAU\_CUSTOMS\_PORT WHERE CODE = ?)

**Query Parameters :**

1) Code: from shipment header [from xml]

Change[3]: set INTLCODE in shipment PortOfDischarge

Get Intl country code for shipment header

### [Go to step #1.1.3](#Step_1_1_3)

[Step #1.1.3:](#_Go_to_step_7) get Intl country code for shipment header

SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE CODE= ? and STATUS ='1'

Query parameter:

1. customsCode : from shipment header[from xml]

Changes[4]: set INTLCODE in shipment VesselNationality

Changes that happens in db[5]: add in maf status =”RCV”

Change[6] : add in agent status “ACP”

For every agent: Get PALicByCustomsLic from agent

### [Go to step #1.1.4](#Step_1_1_4)

[Step #1.1.4:](#_Go_to_step_6)

query[x]: Get PALicByCustomsLic(using searchUtilDao):

SELECT DISTINCT LIC\_NO from TSAU\_AGENT\_LIC\_USER WHERE SHAGT\_NO=?

Query parameter

1. customsNo : from agent [from xml]

Case #1:if query exist

Change[7]:set LIC\_NO in agent LicenceNo (parse it to integer)

Case #2:if anything else

Change[7]: set customsNo in agent LicenceNo (parse it to integer)

Change[8]: set agent status “ACP”

For every bill (agent bill): Get Intl port code using bill header

### [Go to step #1.1.5](#Step_1_1_5)

[**Step #1.1.5:**](#_Go_to_step_5)

Get Intl port code (using msgUtil)

SELECT INTLCODE from TSAU\_CUSTOMS\_PORT WHERE CODE = ?)

**Query Parameters :**

1. code: From bill header [from xml]

Change[9]: set INTLCODE in bill header FinalDest.

Change[10]: set [query [x]](#queryX) in bill header ShipAgentNo

Get Intl country code using bill header

### [Go to step #1.1.6](#Step_1_1_6)

[Step #1.1.6:](#_Go_to_step_4) Get Intl country code using bill header

SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE CODE= ? and STATUS ='1'

Query parameter:

1) CODE : From bill Header [from XML]

Change[11]: set INTLCODE in bill header CountryOfOrigin

Change[12]:set INTLCODE+ ForeignLoadingCode in bill header PortOfLoading

Change[13]:Set Doc status in header “ACP”

For every item in bill set pkg type

### [Go to step #1.1.7](#Step_1_1_7)

[Step #1.1.7:](#_Go_to_step_3)  For every item in bill set PKGType

"SELECT \* from TSAU\_CUSTOMS\_PKGTYPE WHERE CODE=?   
 Query parameter:

1. Code: from item (custom pkg type)[from XML]

Change[14]: set INTLCODE in item PKg type

For every container: set container type

### [Go to step #1.1.8](#Step_1_1_8)

Step # 1.1.8: for every Container

"SELECT \* from TSAU\_CUSTOMS\_CONTAINER WHERE CODE= ?   
 Query parameter:

1)code : From container CustContainerCode (agent container)[From xml].

Return list that contain [CTN\_SIZE,CTN\_SIZE]

Change[15]: set list in ContainerType

Change[16]: set container status “ACP”

Change[17]: set PALic in agent LicenceNo

Change[18]: set quary[x] Container ShipAgentNo

Change[19]: set quary[x]in agent LicenceNo

Add all agint in list

Add list in to maf.

For every route (from manifest)

### [Go to step #1.1.9](#Step_1_1_9)

[Step #1.1.9:](#_Go_to_step_1)

For every route (from manifest)

SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE CODE= customsCode and STATUS ='1'

Query parameter:

1)customsCode : from route .

Change[20]:set INTLCODE in rout Country

Change[21]: set route status “ACP”

## Step #1.2: check if docref

Caes #1: If document ref no exist

### [Go to step #1.2.1](#Step_2_1)

Case #2: if document refno not exist

Set document refno with null

## Step #1.3: delete from custom

case #1: if document ref not null and length > 0

set Doc Ref in header and

### [go to step #1.3.1.](#Step_3_1)

## Go to step #1.4: save manifest from custom

### save header go to [step #1.4.1](#Step_4_1)

### Insert all routs go to [step #1.4.2:](#Step_4_2)

### Insert all agent go [to step #1.4.3.](#Step_4_3)

### Insert all agent container go to [step #1.4.4](#Step_4_4)

### Set UI MODE in header “U”

### Save maf status go to step [#1.4.5](#Step_4_5)

### Save bill from custom [go to #1.4.6](#Step_4_6)

Save Recipient [goto#1.4.7](#step_4_7)

## Go to step #1.5: update linked bill

**Use the below query to select message details from tables**

**tsau\_deo\_header t1 & tsau\_deo\_bills t2**

"select t1.acctid, t1.deo\_docrefno, t2.bl\_seqno from tsau\_deo\_header t1,

tsau\_deo\_bills t2 where t1.acctid = t2.acctid and t1.deo\_docrefno =

t2.deo\_docrefno and maf\_no = '"(mafHdr.getCustomsMafNo()) "' and

maf\_issuedate =

to\_date('"(mafHdr.getCustomsDateStr()).append("','DD/MM/YYYY')"' and

cus\_regportcode = '"(mafHdr.getCustomsRegPortCode()) "'"

**query parameters :**

1. t1. acctid
2. t2. acctid
3. t1.deo\_docrefno
4. t2. deo\_docrefno
5. maf\_no: from header [xml message]
6. maf\_issuedate: from header [xml message]
7. cus\_regportcode: from header [xml message]

### Step #1.5.1: update every affected row

**Use the data returned from previous query to update**

**values in table “tsau\_bl\_header”**

"update tsau\_bl\_header set DEO\_ACCTID = ?, DEO\_DOCREFNO = ?

where acctid = (mafHdr.getAcctID()) and docrefno = (mafHdr.getDocRefNo()) || ?"

Note : used values in second query

DEO\_ACCTID = t1.acctid

DEO\_DOCREFNO = t1.deo\_docrefno

? = t2.bl\_seqno

**query parameters :**

1. DEO\_ACCTID
2. DEO\_DOCREFNO
3. Acctid: from header [xml message]
4. Docrefno: from header [changed from DB in change (21)]

[Step #1.2.1](#_Go_to_step):check if docref exist or not

SELECT docrefno FROM VSAU\_MAF\_HEADER WHERE CUSTOMSNO = ? and CUSTOMSDATE =?   
 and MANIFESTTYPE = ? and CUSTOMS\_REGPORTCODE = ?

Query parameter:

CUSTOMSN: from manifest header [from xml message]

CUSTOMSDATE: from manifest header formatted 'DD/MM/YYYY'[from xml message]

MANIFESTTYPE: from manifest header [from xml message]

CUSTOMS\_REGPORTCODE: from manifest header [from xml message]

Change [22]: set document refno in maf doc ref

[Step #1.3.1](#_go_to_step_8) : set Doc Ref in header and

We delete the message info through three steps.

1) from bill table “tsau\_b1\_header\_base”

Go to Step 3.1.1: Delete Message Information

Use the below query to delete message from table

“delete from tsau\_b1\_header\_base where acctid =’?’and docrefno like ‘%’”

Query Parameters :

1 – account id: [from xml message]

2 – document reference number: from xml message [update from database in [changes [22]](#chang22) ]

case#1: if parameters don’t exist :

Throw an Exception

2) from details table “TSAU\_MAF\_DETAILS\_BASE”

Step 3.1.2: Delete Message Information

Use the below query to delete message from table

“delete from TSAU\_MAF\_DETAILS\_BASE where acctid =’’and docrefno‘’”

Query Parameters :

1 – acctid: [from xml message]

2 – docrefno: [update from database in [changes [22]](#chang22) ]

Case#2: if parameters don’t exist :

throw an exception

3) from header table “TSAU\_MAF\_HEADER\_BASE”

Step3.1.3: Delete Message Information

Use the below query to delete message from table

“delete from TSAU\_MAF\_HEADER\_BASE where acctid =’’and docrefno‘’”

Query Parameters :

1 – acctid: from xml message

2 – docrefno: [update from database in [changes [22]](#chang22) ]

Case #3: if parameters don’t exist :

throw an exception

[**STEP #1.4.1: saving message header information**](#_save_header_go)

1- accept message manifest and message header document reference number

2 - get message header using message manifest

3 - fill the header auditInfoKey with manifest auditInfoKey

CASE #1: check if header UI Mode equals “N”

- get all user info from manifest

- Set header account id with user info (account id) attribute

CASE #1.1: check If accepted document Reference doesn’t exist

[- GO TO step [#4.1.1]](#step_4_1_1)

CASE #1.2: else if document reference exists

- Set header document reference number as the accepted one

- update header account id, create user id, last updated user id with values returned from [manifest user info]

- Add Account ID and Document Ref in a list and send it to [[step #4.1.2]](#Step_4_1_2)

CASE #1.1: If header channel equals “H2H”

CASE #1.1.1: check if header auditInfoKeys exists and header activity details exists and contains word “insert”

[GO TO STEP [#4.1.3]](#Step_4_1_3)

CASE #2: check if message header UI Mode equals “U”

[GO TO [step #4.1.5]](#Step_4_1_5)

CASE #2.1: check if message header Channel equals “H2H”

CASE #2.1.1: check if header auditInfoKeys exists and header activity details exists and contains word “update”

[GO TO [step #4.1.6]](#step_4_1_6)

[Step #1.4.2: insert routs](#_Insert_all_routs)

Insert all routes and their keys

Note: keys are a list (list for each route )contain (

Account ID, Document ref: from header [from XML message (id), (docType)]

, SNO: from route [ from XML message (routeSeqNo)]

)

insert list of route and their keys information in DB:

**Insert into TSAU\_MAF\_ROUTE\_BASE (ACCTID, DOCREFNO, SNO, CTRYCODE, PORTCODE, PORTTYPE, REMARKS, STATUS, ERROR\_CODE, ERROR\_REMARKS)values ( ?,?,?,?,?,?,?,?,?,?)");**

**Query parameter:**

**1) ACCTID: from keys list [from xml (id)]**

**2) DOCREFNO: from keys list [changed from db in (step 1.1.9)]**

**Note: insert accid , Docrefno just once**

**3) SNO: from keys list [From XML message (routeSeqNo)]**

**4) CTRYCODE: from routes [from xml message (countryCode)]**

**5) PORTCODE: from routes [from XML message (foreignPort)]**

**6) PORTTYPE: from routes [from xml message]**

**7) REMARKS: from routes [from xml message]**

**8) STATUS: from routs status [change from db (step 1.1.9)**[**]**](#chang21)

**9) ERROR\_CODE: from routs status [From xml]**

**10) ERROR\_REMARKS: from routs status [from xml]**

[Step #1.4.3: insert agent:](#_Insert_all_agent)

Case #1: if agent not empty

Insert all agent and their keys

Note: keys are a list (list for each route) contain (

Account ID, Document ref: from header [from XML message (id),(docType)]

, SNO [from xml message (routeSeqNo)])

insert List of agent and their keys information in DB:

**insert into TSAU\_MAF\_SHIPAGT\_BASE (ACCTID, DOCREFNO, SNO, SHIPAGENTNO, CUSTOMSAGENTNO, SHIPAGENTTYPE, NOOFBL, NOOFCONTAINER, STATUS, ERROR\_CODE, ERROR\_REMARKS, VIEWABLE\_STATUS, SUBMISSION\_TSTAMP, NOOFLOADPERMIT) values (?,?,?,?,?,?,?,?,?,?,?,?,sysdate,?)")**

**Query parameter:**

**1) ACCTID: from keys list [From xml message (id)]**

**2) DOCREFNO: from keys list[changed from db in** [**(**](#chang22)**step 1.2.1)]**

**Note: insert accid, Docrefno just once**

**3) SNO: from keys list (**start from 1 and increment it by one)

**4) SHIPAGENTNO: [from agent** [**Change[19]**](#chang9)

**5) CUSTOMSAGENTNO: from agent [From xml]**

**6) SHIPAGENTTYPE: from agent if the value exist put 0 if not put 1 [From xml]**

**7)NOOFBL: from agent [From xml]**

**8)NOOFCONTAINER: from agent [From xml]**

**9)STATUS: from status [Fromdb** [**Change[8]]**](#chang8)

**10)ERROR\_CODE: from status [From xml]**

**11)ERROR\_REMARKS: from status [From xml]**

**12)VIEWABLE\_STATUS : from agent [From xml]**

**13)SUBMISSION\_TSTAMP:** sysdate

**14)NOOFLOADPERMIT : from agent [From xml]**

[Step #1.4.4: saving agent containers](#_Insert__all)

Case #1: if agent not empty

For each agent get containers and store them in array list.

Insert all containers and their keys in database

Note: keys are a list (list for each route )contain (

Account ID, Document ref: from header [from XML message]

, SNO: from containers)

**insert list of container and their keys information in DB**:

**insert into VSAU\_MAF\_CONTAINER (ACCTID, DOCREFNO, SNO, SHIPAGENTNO, CONTAINER\_ISO\_TYPE, CONTAINERNO, SEALNO1, SEALNO2, SEALNO3, SHIPMENTTYPE, TAREWT, TAREWTUOM, TEMPERATURESET, TEMPERATUREHI, TEMPERATURELO, TEMPERATUREUOM, MARKSNO, REMARK, GOODSWT, GOODSWTUOM, CONTAINERTYPE, CONTAINERSIZE, STORAGE\_LOC\_CD, CONTAINERHEIGHT, CONTAINERWIDTH, CREATEACCTID, STATUS, ERROR\_CODE, ERROR\_REMARKS, VERIFIEDGROSSMASS, VERIFIER) values ( ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,? )**

**Query parameter:**

**1) ACCTID: from keys list [From xml]**

**2) DOCREFNO: from keys list [changed from db in** [**change[22]]**](#chang22)

**Note: insert accid , Docrefno just once**

**3) SNO: from keys list [From xml]**

**4)SHIPAGENTNO [from db** [**Change[18]**](#chang18)**]**

**5) CONTAINER\_ISO\_TYPE [From xml]**

**6)CONTAINERNO[From xml]**

**7) SEALNO1[From xml]**

**8) SEALNO2[From xml]**

**8) SEALNO3[From xml]**

**9) SHIPMENTTYPE [from db** [**Change[18]]**](#chang18)

**10)TAREWT[From xml]**

**11) TAREWTUOM[From xml]**

**12) TEMPERATURESET[From xml]**

**13) TEMPERATUREHI[From xml]**

**14) TEMPERATURELO[From xml]**

**15)TEMPERATUREUOM[From xml]**

**16)MARKSNO[From xml]**

**17) REMARK[From xml]**

**18) GOODSWT[From xml]**

**19) GOODSWTUOM[From xml]**

**20)CONTAINERTYPE [from db** [**Change[15]]**](#chang15)

**21)CONTAINERSIZE[From xml]**

**22)STORAGE\_LOC\_CD[From xml]**

**23) CONTAINERHEIGHT[From xml]**

**24)CONTAINERWIDTH[From xml]**

**25) CREATEACCTID[From xml]**

**26) STATUS [from db** [**Change[16]]**](#chang16)

**27) ERROR\_CODE[From xml]**

**28)ERROR\_REMARKS[From xml]**

**29) VERIFIEDGROSSMASS[From xml]**

**30)VERIFIER[From xml]**

[step #1.4.5 :save maf satuts](#_Save_maf_status)

inserting document status info in database

accepts : keys list, document status

use the below query to insert

"insert into TSAU\_MAF\_STATUS\_BASE (ACCTID, DOCREFNO, SNO, STATUS, STATUSDATE, STATUSDATE\_AR, OVERALL\_STATUS) values(?,?,?,?,SYSDATE,?,?)"

Query parameter:

1) acctID: from header [xml message]

2) DOCREFNO: from header( xml message ) [update from database in [changes [22]](#chang22) ]

3) SNO: fixed value '1'

4) STATUS: from xml message [update from database in [changes [5]](#change5) ]

5) STATUSDATE: Fixed value SYSDATE

6) STATUSDATE\_AR: from status ( unknown resource )

7) OVERALL\_STATUS: from status ( unknown resource )

[Step #1.4.6: save bill](#_Save_bill_from)

get agents list from message manifest

for every agent in agent list

insert agent bills list info in database

go step [#3.6.1](#step_3_6_1) : insert

check if bill routes exist : [step #3.6.2](#step_3_6_2)

[step #3.6.3](#step_3_6_3) : insert bill header

[step#3.6.4](#step_3_6_4): save item from custom

[step #3.6.5](#step_3_6_5) :save Parties

[Step #3.6.1](#GOstep_3_6_1)

"insert into VSAU\_BL\_HEADER ( ACCTID, DOCREFNO, DOCSTATUS, DOCDATE, COMPLETEIND, MSGFUNCCODE, BLIND, DOCTYPE, NOOFORIGINAL, NOOFDUPLICATE, SINO, CARRIAGECOND, CONTAINERSTATUS, SHIPMENTMOVETYPE, SHIPMENTMOVETYPE2, ETD, ETA, TRANSPORTMODE, CARRIERAGENTCODE, TRADINGPARTNERID, TRADINGPARTNERACCTID, BKDOCREFNO, BOOKINGREFNO, FREIGHTIND, TERMSOFDELIVERY, PLACEOFTERMSOFDELIVERY, OBLNO, HBLNO, FREIGHTCHGPAYABLEAT, BASECURR, PLACEOFISSUE, DATEOFISSUE, SHIPONBOARDDATE, TOTALNOOFPKGS, TOTALPACKAGETYPE, TOTALGROSSWT, TOTALGROSSWTUOM, TOTALMEASUREMENT, TOTALMEASUREMENTUOM, TOTALPPCHARGE, TOTALPPCURCODE, TOTALCCCHARGE, TOTALCCCURCODE, MOTHERVESSELNAME, MOTHERVOYAGENO, PRECARRIAGEVESSELNAME, PRECARRIAGEVOYAGENO, PORTOFLOADING, PORTOFLOADINGTYPE, PORTOFDISCHARGE, FINALDEST, FINALDESTPORTTYPE, PLACEOFDELIVERY, PLACEOFRECEIPT, CREATEUSERID, LASTUPDATEDUSERID, LASTUPDATEDDATE, DOCFORMAT, BILLTYPE, STORAGE\_LOC\_CD, SHIPPINGLINEID, TONSQTY, BARRELSQTY, CONCENTRATELVL, OIL\_UNLD\_PORT\_STA, SHIPAGENTNO, CTYOFORGIN, SHIPMENTYPE, UNLOADDATE, UNLOADDATE\_AR, CONTAINERQTY, LINEITEMQTY, GENERALDESC, REMARKS, OWNERNO, CREATEACCTID, PC\_ACCTID ) values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

**query parameters :**

1. ACCTID : from Manifest header [ from xml message ]
2. DOCREFNO : from bill header routes [changed from db in change[22] ]
3. DOCSTATUS : from bill header routes [changed from db in change[13] ]
4. DOCDATE : from bill header routes [ from xml message ]
5. COMPLETEIND : from bill header routes [ from xml message ]
6. MSGFUNCCODE : from bill header routes [ from xml message ]
7. BLIND : from bill header routes [ from xml message ]
8. DOCTYPE : from bill header routes [ from xml message ]
9. NOOFORIGINAL : from bill header routes [ from xml message ]
10. NOOFDUPLICATE : from bill header routes [ from xml message ]
11. SINO : from bill header routes [ from xml message ]
12. CARRIAGECOND : from bill header routes [ from xml message ]
13. CONTAINERSTATUS : from bill header routes [changed from db in change[16] ]
14. SHIPMENTMOVETYPE : from bill header routes [ from xml message ]
15. SHIPMENTMOVETYPE2 : from bill header routes [ from xml message ]
16. ETD : from bill header routes [ from xml message ]
17. ETA : from bill header routes [ from xml message ]
18. TRANSPORTMODE : from bill header routes [ from xml message ]
19. CARRIERAGENTCODE : from bill header routes [ from xml message ]
20. TRADINGPARTNERID : from bill header routes [ from xml message ]
21. TRADINGPARTNERACCTID : from bill header routes [ from xml message ]
22. BKDOCREFNO : from bill header routes [ from xml message ]
23. BOOKINGREFNO : from bill header routes [ from xml message ]
24. FREIGHTIND : from bill header routes [ from xml message ]
25. TERMSOFDELIVERY : from bill header routes [ from xml message ]
26. PLACEOFTERMSOFDELIVERY : from bill header routes [ from xml message ]
27. OBLNO : from bill header routes[ from xml message ]
28. HBLNO : from bill header routes [ from xml message ]
29. FREIGHTCHGPAYABLEAT : from bill header routes [ from xml message ]
30. BASECURR : from bill header routes [ from xml message ]
31. PLACEOFISSUE : from bill header routes [ from xml message ]
32. DATEOFISSUE : from bill header routes [ from xml message ]
33. SHIPONBOARDDATE : from bill header routes [ from xml message ]
34. TOTALNOOFPKGS : from bill header routes [ from xml message ]
35. TOTALPACKAGETYPE : from bill header routes [ from xml message ]
36. TOTALGROSSWT : from bill header routes [ from xml message ]
37. TOTALGROSSWTUOM : from bill header routes [ from xml message ]
38. TOTALMEASUREMENT : from bill header routes [ from xml message ]
39. TOTALMEASUREMENTUOM : from bill header routes [ from xml message ]
40. TOTALPPCHARGE : from bill header routes [ from xml message ]
41. TOTALPPCURCODE : from bill header routes [ from xml message ]
42. TOTALCCCHARGE : from bill header routes [ from xml message ]
43. TOTALCCCURCODE : from bill header routes [ from xml message ]
44. MOTHERVESSELNAME : from bill header routes [ from xml message ]
45. MOTHERVOYAGENO : from bill header routes [ from xml message ]
46. PRECARRIAGEVESSELNAME : from bill header routes [ from xml message ]
47. PRECARRIAGEVOYAGENO : from bill header routes [ from xml message ]
48. PORTOFLOADING : from bill header routes [changed from db in change[12] ]
49. PORTOFLOADINGTYPE : from bill header routes [ from xml message ]
50. PORTOFDISCHARGE : from bill header routes [changed from db in change[3] ]
51. FINALDEST : from bill header routes [changed from db in change[9] ]
52. FINALDESTPORTTYPE : from bill header routes [ from xml message ]
53. PLACEOFDELIVERY : from bill header routes [ from xml message ]
54. PLACEOFRECEIPT : from bill header routes [ from xml message ]
55. CREATEUSERID : from bill header routes [ from xml message ]
56. LASTUPDATEDUSERID : from bill header routes [ from xml message ]
57. LASTUPDATEDDATE : from bill header routes [ from xml message ]
58. DOCFORMAT : from bill header routes [ from xml message ]
59. BILLTYPE : from bill header routes [ from xml message ]
60. STORAGE\_LOC\_CD : from bill header routes [ from xml message ]
61. SHIPPINGLINEID : from bill header routes [ from xml message ]
62. TONSQTY : from bill header routes [ from xml message ]
63. BARRELSQTY : from bill header routes [ from xml message ]
64. CONCENTRATELVL : from bill header routes [ from xml message ]
65. OIL\_UNLD\_PORT\_STA : from bill header routes [ from xml message ]
66. SHIPAGENTNO : from bill header routes [changed from db in change[18] ]
67. CTYOFORGIN : from bill header routes [ from xml message ]
68. SHIPMENTYPE : from bill header routes [ from xml message ]
69. UNLOADDATE : from bill header routes [ from xml message ]
70. UNLOADDATE\_AR : from bill header routes [ from xml message ]
71. CONTAINERQTY : from bill header routes [ from xml message ]
72. LINEITEMQTY : from bill header routes [ from xml message ]
73. GENERALDESC : from bill header routes [ from xml message ]
74. REMARKS : from bill header routes [ from xml message ]
75. OWNERNO : from bill header routes [ from xml message ]
76. CREATEACCTID : from bill header routes [ from xml message ]
77. PC\_ACCTID : from bill header routes [ from xml message ]

[step #3.6.2](#GOstep_3_6_2) : check if bill routes exist :

**insert bill routes info in database**

"insert into TSAU\_BL\_ROUTE ( BL\_ACCTID, BL\_DOCREFNO,

ROUTE\_SEQ\_NO, COUNTRY\_CODE, PORT\_CODE, PORT\_TYPE, REMARKS ) values (?,?,?,?,?,?,?)"

**query parameters :**

1. BL\_ACCTID : from bill header routes [ xml message ]
2. BL\_DOCREFNO : from bill header routes [ xml message ]
3. ROUTE\_SEQ\_NO : from bill header routes [ xml message ]
4. COUNTRY\_CODE : from bill header routes [changed from db in change[20] ]
5. PORT\_CODE : from bill header routes [changed from db in change[1] ]
6. PORT\_TYPE : from bill header routes [ xml message ]
7. REMARKS : from bill header routes [ xml message ]

[step #3.6.3](#GOstep_3_6_3) : insert bill header

**Insert into TSAU\_MAF\_DETAILS\_BASE (ACCTID, DOCREFNO, SNO, BL\_DOCACCTID, BL\_DOCREFNO) values (?,?,?,?,? )"**

Query parameters:

1. **ACCTID :** from manifest header [ xml message]
2. **DOCREFNO :** from manifest header [ xml message]
3. **SNO :** from bill header [ xml message ]
4. **BL\_DOCACCTID :** from billheader [ xml message ]
5. **BL\_DOCREFNO :** : from bill header routes [changed from db in change[22]]

[STEP #4.1.1: **generate new document sequence number**](#GOTOstep_4_1_1)

- accepts :

- account id as “SAU”

- document type as “MAF”

- length of sequence as “6” integer number

- flag as 108

- boolean oracle attribute as false

- initialize seqnum\_str as “0” string

- initialize seqnum as “-1” integer

**- select sequence number, system date using below query**

- "SELECT SYS\_DATE, SEQNO FROM TBAS\_SEQ\_NO WHERE ACCTID=? And

DOCTYPE=? FOR UPDATE “

**Query parameters :**

**1 – account id “SAU” [staticly inserted]**

**2 – document type “MAF” [staticly inserted]**

CASE #1 : check if there is data returned from the select query, if there is :

- get current date string

- CASE #1.1 : check if current date equal returned system date :

- seqnum\_str = returned sequence number casted to string then incremented

- CASE # 1.2 : else if current date and system date aren’t equal :

- seqnum\_str = “1” string  
 - sequence number = “2” integer

- returned system date equals current date

**- update info in database using below query**

“UPDATE TBAS\_SEQ\_NO SET SEQNO=?, SYS\_DATE=? WHERE ACCTID=? AND DOCTYPE=?"

**Query parameters :**

**1 – sequence number [returned from select query or initialized to-1, or has been set as “2” integer]**

**2 – system date [returned from select query or equals currentdate]**

**3 – account id “Sau” [staticly inserted]**

**4 – document type “MAF” [staticly inserted]**

CASE #2 : else if there is no data returned :

**- insert info in database using below query**

"INSERT into TBAS\_SEQ\_NO (ACCTID,DOCTYPE,SYS\_DATE,SEQNO) VALUES (?,?,?,?)"

**Query Parameters :**

**1 – account id “SAU” [staticly inserted]**

**2 – document tupe “MAF” [staticly inserted]**

**3 – current date [calculated]**

**4-** SEQNO [statically inserted by ‘2’]

- set seqnum\_str as “1” string

- return seqnum\_str

- set times variable as [length of sequence number “6” – returnd seqnum\_str length]  
- iterate on times, while it doesn’t equals 0 yet, decrement times and add “0” string to

seqnum\_str

- get current time, add it to seqnum\_str. And return the result as the new generated document sequence number

**END STEP #4.1.1**

[**STEP #4.1.2: inserting header info in database**](#GOTOstep_4_1_2)

- accepts the message header and list contains (user id, document reference number)

CASE#1 : check If header UI Mode equals “N”

- **Use the below query to insert values into table VSAU\_MAF\_HEADER**

( Insert into VSAU\_MAF\_HEADER (ACCTID, DOCREFNO, MSGSEQNO, DOCDATE, COMPLETEIND, MANIFESTNO, MANIFESTDATE, MANIFESTTYPE, VESSELNAME, VESSELNATIONALITY, VOYAGENO, PORTOFLOADING, PORTOFDISCHARGE, PORTOFDISCHARGETYPE, FINALDEST, ETD, ETA, UNLOADDATE, REMARKS, CREATEUSERID, LASTUPDATEDUSERID, LASTUPDATEDDATE, DELETEFLAG, TRANSPORTMODE, PARTNERACCTID, PARTNERUSERID, DOCFORMAT, SHIPONBOARDDATE, TOTALBL, TOTALCONTAINER, CUSTOMSNO, CUSTOMSDATE, CUSTOMSDATE\_AR, VESSELNAME\_AR, ETD\_AR, ETA\_AR, UNLOADDATE\_AR, CAPTAINNAME, CAPTAINNAME\_AR, TANKER\_GROSS\_WT, TANKER\_NET\_WT,TRANS\_TYPE\_CODE, LLOYDNO, TOTALPASSENGER,CHANNEL,CUSTOMS\_REGPORTCODE) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,TO\_DATE(?,'YYYYMMDDHH24MISS'),TO\_DATE(?,'YYYYMMDDHH24MISS'),?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

**Query Parameters :**

**1 – account id [ from xml message ]**

**2 – document reference number [ from change[22] in database ]**

**3 – header mesasge sequence number [from xml message]**

**4 – header document date [ from xml message ]**

**5 – “N” [staticly inserted]**

**6 – header manifest number [ from xml message ]**

**7 – header manifest date [ from xml message ]**

**8 – header manifest type [ from xml message ]**

**9 – shipment vessel name [ from xml message ]**

**10 – shipment vessel nationality [ from change [4] in database]**

**11- shipment voyage number [ from xml message ]**

**12 – shipment port of loading [ from xml message ]**

**13 – shipment port of discharge [ from change[3] in database ]**

**14 - shipment port of discharge type [ from xml message ]**

**15 – shipment final destination desc [ from xml message ]**

**16 – shipment etdStr [ from xml message ]**

**17 – shipment etdStr [ from xml message ]**

**18 – shipment unload Date [from xml message]**

**19 – header remarks [ from xml message ]**

**20 – header user id [ from xml message ]**

**21 – header last updated user id [ from xml message ]**

**22 – header last updated date [ from xml message ]**

**23 – header deleted flag [ from xml message ]**

**24 – shipment transport code [ from xml message ]**

**25 – header partner account [ from xml message ]**

**26 – header partner user id [ from xml message ]**

**27 – header document format [ from xml message ]**

**28 – shipment ship on board date [ from xml message ]**

**29 – header total bills [ from xml message ]**

**30 – header total containers [ from xml message ]**

**31 – header customs MAF number [ from xml message ]**

**32 – header customs date [ from xml message ]**

**33 – header customs date in arabic [ from xml message ]**

**34 – shipment vessel name in arabic [ from xml message ]**

**35 – shipment ETA in arabic [ from xml message ]**

**36 – shipment ETD in arabic [ from xml message ]**

**37-shipment unload date in arabic**

**38 – shipment captain name [ from xml message ]**

**39 - shipment captain name in arabic [ from xml message ]**

**40 – shipment tanker gross weight [ from xml message ]**

**41 – shipment tanker net weight [ from xml message ]**

**42 – shipment transport unit type code [ from xml message ]**

**43 – shipment vesselLloydNo [ from xml message ]**

**44 – header total passengers [ from xml message ]**

**45 – header channel [ from xml message ]**

**46 – header custom registeration port code [ from xml message ]**   
**END STEP #4.1.2**

[**STEP #4.1.3 : inserting header audit info in database**](#GOTOstep_4_1_3)

- accepts message header and header shipment and list contains (user id, document reference number)

- for arabic language, make 2 objects (subject\_AR, account\_AR) that holds the local information

- for english language, make 2 objects (subject\_EN, account\_EN) that holds the local information

**- use below query to set header audit info sql statement**

[QueryX2]:INSERT INTO VSAU\_MAF\_HEADER ( ACCTID, DOCREFNO, MSGSEQNO, DOCDATE, COMPLETEIND, MANIFESTNO, MANIFESTDATE, MANIFESTTYPE, VESSELNAME, VESSELNATIONALITY, VOYAGENO, PORTOFLOADING, PORTOFDISCHARGE, PORTOFDISCHARGETYPE, FINALDEST, ETD, ETA, UNLOADDATE, REMARKS, CREATEUSERID, LASTUPDATEDUSERID, LASTUPDATEDDATE, DELETEFLAG, TRANSPORTMODE, PARTNERACCTID, PARTNERUSERID, DOCFORMAT, SHIPONBOARDDATE, TOTALBL, TOTALCONTAINER, CUSTOMSNO, CUSTOMSDATE, CUSTOMSDATE\_AR, VESSELNAME\_AR, ETD\_AR, ETA\_AR, UNLOADDATE\_AR, CAPTAINNAME, CAPTAINNAME\_AR, TANKER\_GROSS\_WT, TANKER\_NET\_WT, TRANS\_TYPE\_CODE , LLOYDNO, TOTALPASSENGER,CHANNEL,CUSTOMS\_REGPORTCODE ) VALUES ( ?, ? , ?,?, ? , ? , ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ? ,?, ?, ?, ? , ? , ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

**Query Parameters :**

**1 – header account id [ from xml message ]**

**2-header document reference number [ from change[22] in database ]**

**3-header message sequence number [ from xml message ]**

**4-header document date [ from xml message ]**

**5-”N” [staticly inserted]**

**6-header manifest number [ from xml message ]**

**7-header manifest date [ from xml message ]**

**8-header manifest type [ from xml message ]**

**9-shipment vessel nationality [ ffrom change[4] in database ]**

**10-shipment voyage number [ from xml message ]**

**11-shipment port of loading [ from xml message ]**

**12-shipment port of discharge [ from change[3] in database ]**

**13-shipment port of discharge type [ from xml message ]**

**14-shipment final destination desc [ from xml message ]**

**15-shipment ETD “estimated time of departure” [ from xml message ]**

**16- shipment ETA “estimated time of arrival” [ from xml message ]**

**17-shipment unload date [ from xml message ]**

**18-header remarks [ from xml message ]**

**19-header create user id [ from xml message ]**

**20-header Last updated user id [ from xml message ]**

**21-header Last updated date [ from xml message ]**

**22-header delete flag [ from xml message ]**

**23-shipment transport mode [ from xml message ]**

**24-header partner Account id [ from xml message ]**

**25-header partner user id [ from xml message ]**

**26-header Document format [ from xml message ]**

**27-shipment ship on board date [ from xml message ]**

**28-header total bills [ from xml message ]**

**29-header total container [ from xml message ]**

**30-header customs Maf Number [ from xml message ]**

**31-header customs date [ from xml message ]**

**32-header customs Date in arabic [ from xml message ]**

**33-shipment vessel name in arabic [ from xml message ]**

**34-shipment Etd in arabic [ from xml message ]**

**35-shipment Eta in arabic [ from xml message ]**

**36-shipment unload date in arabic [ from xml message ]**

**37-shipment captain name [ from xml message ]**

**38-shipment captain name in arabic [ from xml message ]**

**39-shipment tanker gross weight [ from xml message ]**

**40-shipment tanker nett weight [ from xml message ]**

**41-shipment transport unit type code [ from xml message ]**

**42-shipment Vessel Lloyd No. [ from xml message ]**

**43-header total passengers [ from xml message ]**

**44-header channel [ from change[2] in databse ]**

**45-header customs port code [ from change[1] in database ]**

- make new AuditInfo object

- add the previous query and it’s parameters as the object sql statement

- get port using the following query

" SELECT DISTINCT CODE FROM TSAU\_CUSTOMS\_PORT WHERE INTLCODE = ?

AND TYPE = ? "

**query parameters :**

**1 – headers custom registeration port code [from change[1] in database]**

**2 – port type “1” [staticly inserted]**

- insert auditInfo object port code with returned port

- insert auditInfo :

- activity attribute as account\_EN

- activity\_ar attribute as account\_AR

- remarks attribute as subject\_En

- remarks\_ar attribute as subject\_AR

- activity done by attribute as header audit info key, acocunt by attribute

- activity date attribute as current date

- sau\_account id as header account id

- sau\_user\_id as header audit info user id

- document reference number as header document reference number

- module as “9” integer number

- current date as calender instance current time

- add audit info log to database using below query

**-** "INSERT INTO TSAU\_AUDTRL\_LOG (DOCREFNO, ACCTID, USERID, MODULE, PORT, ACTIVITY,REMARKS, SQL\_QRY1, SQL\_QRY2, SQL\_QRY3, ACT\_DATE, ACT\_BY, ACTIVITY\_AR, REMARKS\_AR) VALUES (?,?,?,?,?,?,?,?,?,?,

TO\_DATE(?, 'yyyy/DD/MM HH24:MI:SS'),?,?,?)")

**query parameters :**

**1 - audit info document reference number [from change[22] in database]**

**2 - audit info account id [ from xml message ]**

**3 - audit info user id [ from xml message ]**

**4 - audit info module [“9” integer staticly inserted]**

**5 - audit info port [from previous select query database]**

**6 - audit info activity [ as local english]**

**7 - audit info remarks [as local english ]**

**8 - audit info sql statement [ from sql query in** [**[QueryX3]**](#QueryX2)**]**

**9 – null [ staticly inserted ]**

**10 – null [ staticly inserted ]**

**11 - audit info activity date [ calculated “current time” ]**

**12 - audit info activity done by [from message xml]**

**13 - audit info activity\_ar [ local arabic ]**

**14 - audit info remarks\_ar [ local arabic ]**

**END STEP #4.1.3**

[**STEP #4.1.5 : update header information**](#GOTOstep_4_1_5)

**Use the below query to Update header information**

(UPDATE VSAU\_MAF\_HEADER SET CUSTOMS\_REGPORTCODE = ?, COMPLETEIND = ?, MANIFESTNO = ?, MANIFESTDATE = ?, MANIFESTTYPE = ?, VESSELNAME = ?, VESSELNATIONALITY = ?, VOYAGENO = ?, PORTOFLOADING = ?, PORTOFDISCHARGE = ?, PORTOFDISCHARGETYPE = ?, FINALDEST = ?, ETD = ?, ETA = ?, UNLOADDATE = ? , REMARKS = ?, CREATEUSERID =?, LASTUPDATEDUSERID = ?, LASTUPDATEDDATE = ?, DELETEFLAG = ?, TRANSPORTMODE = ? , PARTNERACCTID = ?, PARTNERUSERID = ?, DOCFORMAT = ?, SHIPONBOARDDATE = ?, TOTALBL = ?, TOTALCONTAINER = ?, CUSTOMSNO = ?, CUSTOMSDATE = ?, CUSTOMSDATE\_AR = ?, VESSELNAME\_AR = ?, ETD\_AR = ?, ETA\_AR = ?, UNLOADDATE\_AR = ?, CAPTAINNAME = ?, CAPTAINNAME\_AR = ?, TANKER\_GROSS\_WT = ?, TANKER\_NET\_WT = ?, TRANS\_TYPE\_CODE = ?, LLOYDNO = ?, MSGSEQNO=? WHERE ACCTID = ? and DOCREFNO = ? )

**Query Parameters :**

**1-header customs registeration port code [ from change[1] in database]**

**2-header complete ind [ from xml message ]**

**3-header manifest number [ from xml message ]**

**4-header manifest date [ from xml message ]**

**5-header manifest type [ from xml message ]**

**6-shipment vessel name [ from xml message ]**

**7-shipment vessel nationality [ from change[4] in database ]**

**8-shipment voyage Number [ from xml message ]**

**9-shipment port of loading [ from xml message ]**

**10-shipment port of discharge [ from change[3] in database]**

**11-shipment port of discharge type [ from xml message ]**

**12-shipment final destination desc [ from xml message ]**

**13-shipment EtdStr [ from xml message ]**

**14-shipment EtaStr [ from xml message ]**

**15-shipment unload date [ from xml message ]**

**16-header remarks [ from xml message ]**

**17-header create user id [ from xml message ]**

**18-header last updated user id [ from xml message ]**

**19-header last updated date [ from xml message ]**

**20-header delete flag [ from xml message ]**

**21-shipment transport mode [ from xml message ]**

**22-header partner Acctount id [ from xml message ]**

**23-header partner user id [ from xml message ]**

**24-header document format [ from xml message ]**

**25-shipment ship on board date [ from xml message ]**

**26-header total Bills [ from xml message ]**

**27-header total containers [ from xml message ]**

**28-header customsMafNumber [ from xml message ]**

**29-header customs date [ from xml message ]**

**30-header customs date in arabic AR [ from xml message ]**

**31-shipment vessel name in arabic [ from xml message ]**

**32-shipment Etd in arabic [ from xml message ]**

**33-shipment Eta in arabic [ from xml message ]**

**34-shipment unload date in arabic [ from xml message ]**

**35-shipment captain name [ from xml message ]**

**36-shipment captain name in arabic[ from xml message ]**

**37-shipment tanker gross weight [ from xml message ]**

**38-shipment.getTankerNettWeight [ from xml message ]**

**39-shipment transport unit type code [ from xml message ]**

**40-shipment vesselLloydNo [ from xml message ]**

**41-header Message Sequence Number [ from xml message ]**

**42-header account id [ from xml message ]**

**43-header document reference number [ from xml message ]**

**END STEP #4.1.5**

[**STEP #4.1.6 : update header audit info**](#GOTOstep_4_1_6)

**Use the below query to Update header information**

[Quary[x3]:](#queryX3)(UPDATE VSAU\_MAF\_HEADER SET CUSTOMS\_REGPORTCODE = ?, COMPLETEIND = ?, MANIFESTNO = ?, MANIFESTDATE = ?, MANIFESTTYPE = ?, VESSELNAME = ?, VESSELNATIONALITY = ?, VOYAGENO = ?, PORTOFLOADING = ?, PORTOFDISCHARGE = ?, PORTOFDISCHARGETYPE = ?, FINALDEST = ?, ETD = ?, ETA = ?, UNLOADDATE = ? , REMARKS = ?, CREATEUSERID =?, LASTUPDATEDUSERID = ?, LASTUPDATEDDATE = ?, DELETEFLAG = ?, TRANSPORTMODE = ? , PARTNERACCTID = ?, PARTNERUSERID = ?, DOCFORMAT = ?, SHIPONBOARDDATE = ?, TOTALBL = ?, TOTALCONTAINER = ?, CUSTOMSNO = ?, CUSTOMSDATE = ?, CUSTOMSDATE\_AR = ?, VESSELNAME\_AR = ?, ETD\_AR = ?, ETA\_AR = ?, UNLOADDATE\_AR = ?, CAPTAINNAME = ?, CAPTAINNAME\_AR = ?, TANKER\_GROSS\_WT = ?, TANKER\_NET\_WT = ?, TRANS\_TYPE\_CODE = ?, LLOYDNO = ?, MSGSEQNO=? WHERE ACCTID = ? and DOCREFNO = ? )

**Query Parameters :**

**1-header customs registeration port code [ from change[1] in database message ]**

**2-header complete ind [ from xml message ]**

**3-header manifest number [ from xml message ]**

**4-header manifest date [ from xml message ]**

**5-header manifest type [ from xml message ]**

**6-shipment vessel name [ from xml message ]**

**7-shipment vessel nationality [ from change[4] in database ]**

**8-shipment voyage Number [ from xml message ]**

**9-shipment port of loading [ from xml message ]**

**10-shipment port of discharge [ from change[3] in database]**

**11-shipment port of discharge type [ from xml message ]**

**12-shipment final destination desc [ from xml message ]**

**13-shipment EtdStr [ from xml message ]**

**14-shipment EtaStr [ from xml message ]**

**15-shipment unload date [ from xml message ]**

**16-header remarks [ from xml message ]**

**17-header create user id [ from xml message ]**

**18-header last updated user id [ from xml message ]**

**19-header last updated date [ from xml message ]**

**20-header delete flag [ from xml message ]**

**21-shipment transport mode [ from xml message ]**

**22-header partner Acctount id [ from xml message ]**

**23-header partner user id [ from xml message ]**

**24-header document format [ from xml message ]**

**25-shipment ship on board date [ from xml message ]**

**26-header total Bills [ from xml message ]**

**27-header total containers [ from xml message ]**

**28-header customsMafNumber [ from xml message ]**

**29-header customs date [ from xml message ]**

**30-header customs date in arabic AR [ from xml message ]**

**31-shipment vessel name in arabic [ from xml message ]**

**32-shipment Etd in arabic [ from xml message ]**

**33-shipment Eta in arabic [ from xml message ]**

**34-shipment unload date in arabic**

**35-shipment captain name [ from xml message ]**

**36-shipment captain name in arabic[ from xml message ]**

**37-shipment tanker gross weight [ from xml message ]**

**38-shipment.getTankerNettWeight [ from xml message ]**

**39-shipment transport unit type code [ from xml message ]**

**40-shipment vesselLloydNo [ from xml message ]**

**41-header Message Sequence Number [ from xml message ]**

**42-header account id [ from xml message ]**

**43-header document reference number** **[ from xml message ]**

- make new AuditInfo object

- add the previous query and it’s parameters as the object sql statement

- get port using the following query

" SELECT DISTINCT CODE FROM TSAU\_CUSTOMS\_PORT WHERE INTLCODE = ?

AND TYPE = ? "

**query parameters :**

**1 – headers custom registeration port code [ from change[1] in database]**

**2 – port type “1” [staticly inserted]**

- insert auditInfo object port code with returned port

- insert auditInfo :

- activity attribute as account\_EN

- activity\_ar attribute as account\_AR

- remarks attribute as subject\_En

- remarks\_ar attribute as subject\_AR

- activity date attribute as current date

- sau\_account id as header account id

- sau\_user\_id as header last updated id

- document reference number as header document reference number

- module as “9” integer number

- current date as calender instance current time

**- add audit info log to database using query**

INSERT INTO TSAU\_AUDTRL\_LOG (DOCREFNO, ACCTID, USERID, MODULE, PORT, ACTIVITY,REMARKS, SQL\_QRY1, SQL\_QRY2, SQL\_QRY3, ACT\_DATE, ACT\_BY, ACTIVITY\_AR, REMARKS\_AR) VALUES (?,?,?,?,?,?,?,?,?,?,TO\_DATE(?, 'yyyy/DD/MM HH24:MI:SS'),?,?,?));

**query parameters :**

**1 - audit info document reference number [ from change[22] in database ]**

**2 - audit info account id [ from xml message ]**

**3 - audit info user id [ from xml message ]**

**4 - audit info module [ “9” integer staticly inserted]**

**5 - audit info port [ from previous select query in database]**

**6 - audit info activity [ local english]**

**7 - audit info remarks [ local english]**

**8 - audit info sql statement [ from sql statement in** [**quary[x3]**](#QuaryX3)**]**

**9 – null [ staticly inserted]**

**10 – null [ staticly inserted]**

**11 - audit info activity date [ calculated “current date” ]**

**12 - audit info activity done by [ from xml message ]**

**13 - audit info activity\_ar [ local arabic]**

**14 - audit info remarks\_ar [ local arabic]**

**END STEP #4.1.6**

[step #3.6.4](#GOstep_3_6_4) : saving bill items in database

use the below query to Insert bill items details in data base

**insert into VSAU\_BL\_DETAIL (ACCTID, DOCREFNO, SNO, NOOFPKGS, PKGTYPE, GROSSWEIGHT, GROSSWEIGHTUOM, MEASUREMENT, MEASUREMENTUOM, PRODUCTITEM, PRODUCTITEMDESC, ITEMQTY, SHIPPINGTERM, SUPPLIERID, CONTAINERNO, CNTRSEQNOREF, STCQTY, STCUOM, CUSTOMSGOODSTYPE, PATARIFFCODE, STORAGE\_LOC\_CD, DANGER\_STATUS\_CD, IMDG\_CD, PRODUCTITEMDESC\_AR, ACTUAL\_QTY, ACTUAL\_WT, QTY\_BALANCE, WT\_BALANCE, ITEM\_STATUS\_CD, MARKSANDNO, MARKSANDNO\_AR) values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");**

**query parameters :**

1. ACCTID: from BillHeader [ xml message message]
2. DOCREFNO: from BillHeader [changed from db in change (22)]
3. SNO: 1
4. NOOFPKGS: 0
5. PKGTYPE: from BillHeader [changed from DB in change (14)]
6. GROSSWEIGHT: from BillHeader [xml message wieghtUOM]
7. GROSSWEIGHTUOM: from BillHeader [xml message wieghtUOMDesc]
8. MEASUREMENT: from BillHeader [xml message measurement]
9. MEASUREMENTUOM: from BillHeader [xml message measurementUOM]
10. PRODUCTITEM: from BillHeader [xml message productCode]
11. PRODUCTITEMDESC: from [xml message productCodeDesc]
12. ITEMQTY: 0
13. SHIPPINGTERM: from BillHeader [xml message shippingTerm]
14. SUPPLIERID: from BillHeader [xml message supplierId]
15. CONTAINERNO: from BillHeader [xml message containerNo]
16. CNTRSEQNOREF: from BillHeader [xml message cntrSeqnoRef]
17. STCQTY: 0
18. STCUOM: from BillHeader [xml message STC\_UOM]
19. CUSTOMSGOODSTYPE: from BillHeader [xml message goodsType]
20. PATARIFFCODE: from BillHeader [xml message tariff]
21. STORAGE\_LOC\_CD: from BillHeader [xml message storageLoc]
22. DANGER\_STATUS\_CD: 1
23. IMDG\_CD: from BillHeader [xml message IMDGClass]
24. PRODUCTITEMDESC\_AR: from BillHeader [xml message productDesc\_AR]
25. ACTUAL\_QTY: from BillHeader [xml message actualQuantity]
26. ACTUAL\_WT: from BillHeader [xml message actualWeight]
27. QTY\_BALANCE: from BillHeader [xml message qtyBalance]
28. WT\_BALANCE: from BillHeader [xml message wieghtBalance]
29. ITEM\_STATUS\_CD: 0
30. MARKSANDNO: from BillHeader [XML message markNo]
31. MARKSANDNO\_AR: from BillHeader [XML message markNo\_AR]

[step 3.6.5:](#GOstep_3_6_5)

method : SaveParties

accepts : one bill

step 3.6.5.1

Get information like parties, getAccountID, DocRefNo and getDeletedParties from bill

step 3.6.5.2

preparePartyDelKeys(getAccountID, DocRefNo, getDeletedParties )

Return keys about parties to delete it

Case1#

if party UI mode is equal 'N' : add keys to keyListN

Case2#

if party UI Mode is not equal 'N' : add keys to keyListU

Case3#:

if keyListU exists : update in database

"update VSAU\_BL\_PARTY set NAME = ?, ADDRESS = ?, POSTALCODE = ?, CONTACTPERSON = ?, TELNO = ?, FAXNO = ?, EMAIL = ?, ABBRNAME = ?, CRNO = ?, GSTNO = ?, PARTYID = ?, POBOX = ?, NAME\_AR =?, IMPR\_NBR=? where ACCTID=? and DOCREFNO=? and PARTYTYPE=?"

Query Parameters :

1 - NAME: from party ( unknown resource )

2 - ADDRESS: from party ( unknown resource )

3 - POSTALCODE: POSTALCODE: from party ( xml message)

4 - CONTACTPERSON: from party ( unknown resource )

5 - TELNO: POSTALCODE: from party ( xml message)

6 - FAXNO: POSTALCODE: from party ( xml message)

7 - EMAIL: POSTALCODE: from party ( xml message)

8 - ABBRNAME: from party ( unknown resource )

9 - CRNO: from party ( unknown resource )

10 - GSTNO: from party ( unknown resource )

11 - PARTYID: POSTALCODE: from party ( xml message)

12 - POBOX: POSTALCODE: from party ( xml message)

13 - NAME\_AR: POSTALCODE: from party ( xml message)

14 - IMPR\_NBR:POSTALCODE: from party ( xml message)

15 – ACCTID: from bill ( xml message )

16 – DOCREFNO: from bill ( xml message ) [update from database in changes [21] ]

17 - PARTYTYPE: from bill ( xml message )

Case4#:

if keyListN exists : insert in database

insert into VSAU\_BL\_PARTY "ACCTID, DOCREFNO, PARTYTYPE, NAME, ADDRESS, POSTALCODE, CONTACTPERSON, TELNO, FAXNO, EMAIL, ABBRNAME, CRNO, GSTNO, PARTYID, POBOX, NAME\_AR,IMPR\_NBR" values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

Query Parameters :

1 – ACCTID: from bill( xml message )

2 – DOCREFNO: from bill( xml message ) [update from database in changes [22] ]

3 - PARTYTYPE: from bill ( xml message )

4 - NAME: from party ( unknown resource )

5 - ADDRESS: from party ( unknown resource )

6 - POSTALCODE: from party ( xml message)

7 - CONTACTPERSON: from party ( unknown resource )

8 - TELNO: from party ( xml message)

9 - FAXNO: from party ( xml message)

10 - EMAIL: from party ( xml message)

11 - ABBRNAME: from party ( unknown resource )

12 - CRNO: from party ( unknown resource )

13 - GSTNO: from party ( unknown resource )

14 - PARTYID: from party ( xml message )

15 - POBOX: from party ( xml message)

16 - NAME\_AR: from party ( xml message)

17 - IMPR\_NBR: from party ( xml message)

case5#:

if partiesD exists : delete in database

"delete from VSAU\_BL\_PARTY where ACCTID=? and DOCREFNO=? and PARTYTYPE=?";

Query Parameters :

1 – acctid: from bill( xml message )

2 – docrefno: from bill( xml message ) [update from database in changes [22] ]

3 - PARTYTYPE: from bill ( xml message )

[Step #1.4.7:](#goStep_4_7) Deal with recipients info through:-

* get message header using message manifest
* get list of recipients using message manifest
* for every recipient account in recipient list

case #1: check if recipient port does not exist

set the port as header shipment port of discharge

step #4.7.1: delete recipient records from database

use this query to delete recipients records from database

**"DELETE FROM TSAU\_MAF\_RECIPIENT WHERE ACCTID = ? AND**

**DOCREFNO = ?"**

Query Parameters:

* + - * ACCTID: from header [XML Message]
* DOCREFNO: from [changed from DB in change (21)]

Note: list of recipients from message header

step #4.7.2: insert recipient records in database

use this query to add recipients records in database

**"insert into TSAU\_MAF\_RECIPIENT**

**(ACCTID, DOCREFNO, SNO, RECIPIENTROLE, RECIPIENTUSERID, DATERECEIVED, RECIPIENTINTLPORT, RECIPIENTCUSTOMSPORT, RECIPIENTNAME, RECIPIENTACCTID, RECIPIENTNAME\_AR) values (?,?,?,?,?,SYSDATE,?,?,?,?,?)"**

Query Parameters:

* ACCTID: from header [XML Message]
* DOCREFNO: from header [changed from DB in change (21)]
* SNO: from header [XML Message]
* RECIPIENTROLE: from header [XML Message]
* RECIPIENTUSERID: from header [XML Message]
* DATERECEIVED: [static value=SYSDATE]
* RECIPIENTINTLPORT: from header [XML Message]
* RECIPIENTCUSTOMSPORT: from header [XML Message]
* RECIPIENTNAME: from header [XML Message]
* RECIPIENTACCTID: from header [XML Message]
* RECIPIENTNAME\_AR: from header [XML Message]

Step 4.7.2.1: check the role of recipient account for every account in recipient list received

case #1: if role “CS”

add this account to “customs” list

case #2: if role “PA”

add this account as “port authority”

set the account port as recipient port of discharge

Step 4.7.3:

case#1: check if header channel equals “H2H”

set the header “getAuditInfoKey()”

with account Id, document reference, user Id

add the header to message manifes

# STEP #2: inserting in message history

Use this query to insert those values in table **TSAU\_DOC\_MSGHISTORY**

**"INSERT INTO TSAU\_DOC\_MSGHISTORY (ACCTID,DOCREFNO,MSGID,DOCTYPE,MSGTYPE,INOUTFLAG,HISTORYDATE,ORIGINAL\_MSG\_TYPE,STATUS,FILLER4,FILLER1,FILLER2) values(?,?,?,?,?,?,?,?,?,?,?,?)"**

**Query parameters:**

**1 - header account id [from xml message]**

**2 - header document reference number [change [22] in database]**

**3 - message id [from xml message]**

**4 - document type [static “MAF”]**

**5 - message type [from xml message]**

**6 - history in and out “O”[staticly inserted]**

**7 – current time stamp [calculated]**

**8 - history original message type [unknown]**

**9 - history status [unknown]**

**10 - maf user info user id [from xml message]**

**11 - history filler1[unknown]**

**12 - history filler2 [unknown]**

# Step #3 :Send Msg:

## STEP #3.1: set Maf for xml:

get header from message manifest

get header shipment

key = shipment port of discharge "string" + shipment port of discharge type "casted to string"

CASE #1 check if value of key is hashed in customsLocalPortCode hashmap

return the value of key (local port)

CASE #2 if value is null: get the local port code using below query

**" SELECT CODE from TSAU\_CUSTOMS\_PORT WHERE INTLCODE = ? AND**

**TYPE= ?"**

Query Parameters:

1 - shipment port of discharge [from change [3] in database]

2 - shipment port type [from xml message]

put the returned value in key in hashmap

return local port

set shipment customs port of discharge as returned local port

registration port = header customs registration port

CASE #3: check if registration port exists

set registration port as shipment customs port of discharge

CASE #4:

key = header customs registeration port "string" + port type "which is 1" "casted to

string"

CASE #4.1 check if value of key is cashed in customsLocalPortCode hashmap

return the value of key (registeration port)

CASE #4.2 if value is null*:*

get the registeration port code using below query

**" SELECT CODE from TSAU\_CUSTOMS\_PORT WHERE INTLCODE = ?**

**AND TYPE= ? "**

Query Parameters:

1 - header customs registeration port [from change[1] in database]

2 - port type "1 integer" ] [staticly inserted]

put the returned value in key in customsLocalPortCode hashmap

return registeration port

set header customs registeration port as returned registration port

set custNation = 0

set customsCode to -99 as initial value

key = shipment vessel nationality

CASE #5 check if value of key is hashed in customsCtryCode hashmap

return the value of key (customs nationality code)

CASE #6 if value is null:

get the custom nationality code using below query

**"SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE INTLCODE = ? and**

**STATUS = '1'"**

Query Parameters:

1 - shipment vessel nationality [from change[1] in database]

2 – status “1 integer” [staticly inserted]

get the result of "CODE" in the query

put the returned value in key in customsCtryCode hashmap

return custom nationality code

set shipment vessel nationality as returned custom nationality code

get agent list using message manifest

set country code = 0 as initial value

iterate over agent list for every agent:

get agent bills list

get agent containers list

iterate over containers list for every container:

set code to -1 as initial value

key = container size "casted to string" + container type

CASE #7 check if value of key is cashed in customsCntrCode hashmap

return the value of key (customs container code)

CASE #8 if value is null:

get the customs container code using below query

**"SELECT \* from TSAU\_CUSTOMS\_CONTAINER WHERE**

**CTN\_SIZE = ? AND CTN\_TYPE = ? "**

Query parameters:

1 - container size [from xml message]

2 - container type [change [15] in database]

get the result of "CODE" in the query

put the returned value in key in customsCtryCode hashmap

return customs container code

CASE #8.1: check if calue of code is still -1, if true:

get the customs container code using below query

**"SELECT \* from TSAU\_CUSTOMS\_CONTAINER WHERE**

**CTN\_SIZE = ? ORDER BY CODE**

Query parameters:

1 - container size [from xml message]

put the returned value in key in customsCtryCode hashmap

return customs container code

set container customs container code as returned customs container code

set code to -1.0D as initial value

key = container size "casted to string" + container type

CASE #9 check if value of key is cashed in customsCntrTareWeight hashmap

return the value of key (container minimun tare weight)

CASE #10 if value is null :

get the customs container code using below query

**"SELECT mintarewgt from TSAU\_CUSTOMS\_CONTAINER WHERE**

**CTN\_SIZE = ? AND CTN\_TYPE = ? "**

Query parameters :

1 - container size [from xml message]

2 - container type [from change[15] in database]

get the result of "mintarewgt" in the query

put the returned value in key in customsCntrTareWeight hashmap

return container minimun tare weight

CASE #10.1 : check if calue of code is still -1.0D, if true :

get the customs container code using below query

**"SELECT mintarewgt from TSAU\_CUSTOMS\_CONTAINER**

**WHERE CTN\_SIZE = ? ORDER BY CODE**

Query parameter:

1 - container size [from xml message]

put the returned value in key in customsCntrTareWeight

hashmap

return container min tare weight

CASE #11: check if returned container minimum tare weight > container tare

weight, if true:

set container tare weight as the returned container minimum tare weight

CASE #12: check if container shipment type equals 8 :

set container goodsWT as container tare weight

iterate over bills list for every bill:

get bill header

set customsCode to -99 as initial value

key = bill header country of origin

CASE #13 check if value of key is cashed in customsCtryCode hashmap

return the value of key (customs nationality code)

CASE #14 if value is null:

get the customs code using below query

**"SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE**

**INTLCODE = ? and STATUS ='1' "**

Query parameters:

1 - bill header country of origin [from change [11] in database]

get the result of "CODE" in the query

put the returned value in key in customsCtryCode hashmap

return customs code

set bill header customs country code as returned customs code

CASE #15 : check if bill header final destination exists and starts with "SA" :

key = bill header final destination "string" + bill header final destination

type "casted to string"

CASE #15.1: check if value of key is cashed in customsLocalPortCode hashmap

return the value of key (local port)

CASE #15.2 if value is null :

get the local port code using below query

**" SELECT CODE from TSAU\_CUSTOMS\_PORT WHERE**

**INTLCODE = ? AND TYPE= ? "**

Query parameters:

1 - bill header final destination [from change [9] in database]

2 - bill header final destination type [from xml message]

put the returned value in key in hashmap

return local port

CASE #15.3 : check if returned local port exists, if true :

set bill header final destination code as returned customs port

code

CASE #15.4 : else if it doesn't exist

set bill header final destination code as 0

Case #16 : else if bill header final destination doesn't exist :

set bill header final destination code as 0

set bill header foreign loading code as bill header port of loading "starting from

index 2 in the string"

get item list of bill

iterate over item list for every item :

key = bill header country of origin

CASE #17: check if value of key is hashed in customsPkgType hashmap

return the value of key (custom package type)

CASE #18 if value is null:

get the custom package type using below query

**"SELECT \* from TSAU\_CUSTOMS\_PKGTYPE WHERE**

**INTLCODE = ? "**

Query parameter:

1 - bill header country of origin [from change [9] in database]

get the result of "CODE" in the query

put the returned value in key in customsPkgType hashmap

return custom package type

set item customs package type as returned custom package type

calculate bill good weight [STEP #3.1.1]

### STEP #3.1.1: (calculate total bill goods weight):

set total weight to 0.0D as initial value

CASE #1 : check if header shipment type equal 8

iterate over bill items for every item:

CASE #1.1: check if item good type equal 11

get item container sequence number reference

get container from manifest using returned container number reference

CASE #1.2: check if container exists, if true :

set item weight as returned container tate weight

add item weight to total bill weight

set header total gross weight as bill total weight

END STEP[#3.1.1]

get bill route info list

iterate over bill route info list for every bill route :

CASE #19: check if bill route info country code exists, if true:

set customs code to -99 as initial value

key = bill Rroute info country code

CASE #19.1 check if value of key is hashed in customsCtryCode hashmap

return the value of key (customs code)

CASE #19.2 if value is null:

get the local port code using below query

**"SELECT \* from from TSAU\_CUSTOMS\_COUNTRY**

**WHERE INTLCODE = ? AND STATUS = 1 "**

Query parameters:

1 - bill Rroute info country code [from xml message]

get the result of "CODE" in the query

put the returned value in key in

customsCtryCode hashmap

return customs code

CASE #19.3: check if returned customs code not equal -99, if true:

set bill route info country code as returned customs code

set bill route info code as bill route info port code "starting from index 2 in the string"

get route list from message manifest

iterate over route list for every route:

set customsCode to -99 as initial value

key = route country

CASE #20 check if value of key is hashed in customsCtryCode hashmap

return the value of key (customs code)

CASE #21 if value is null:

get the customs code using below query

**"SELECT \* from TSAU\_CUSTOMS\_COUNTRY WHERE INTLCODE = ?**

**and STATUS ='1' "**

Query parameters:

1 - route country [from change[20] in database]

2 – status “1 integer” [staticly inserted]

get the result of "CODE" in the query

put the returned value in key in customsCtryCode hashmap

return customs code

set route customs country code as returned customs code

get next message id using below query

**"select MSGID\_SEQ.NEXTVAL from DUAL"**

- construct the message id using "SAU" and returned message sequence number from previous select query

- set message id as "SAU" + current time with format "yyyyMMdd "

get result of equation = 6 - sequence number length

iterate over result for every increment

append "0" to message id

append sequence number to message id

return message id

## STEP #3.2: msgTransformer

Accepts: manifest and message type

### Step #3.2.1: Method: create SEASUB message

Accepts: manifest message

Check the changes in our base manifest

Case1# if submit type of manifest equal '0' and check header available is true:

set carrier maf number in seaMafHeader

set carrier maf date in seaMafHeader

set vessel name in seaMafHeader

set vessel name\_ar in seaMafHeader

set total number of passengers in seaMafHeader

set vessel nationality in seaMafHeader

set voyage number in seaMafHeader

set transport unit type code in seaMafHeader

set VesselLloydNo in seaMafHeader

set port code and port of discharge in seaMafHeader

set ETA in seaMafHeader

set discharge date in seaMafHeader

set marine agent number in seaMafHeader

set remarks in seaMafHeader

set total NOBL in seaMafHeader

set total NOContainers in seaMafHeader

Case1.1# if (submit type of manifest equal '1' and check vesselAgent is true) or *submit type of manifest equal '0':*

set agent number in blAssignment

set number of bill in blAssignment

set NoOfContainers in blAssignment

add blAssignment to seasub

#### step 3.2.1.1

declar iterator about Container then:

set container sequence number in msg container

set shipping agent number in msg container

set container type in msg container

set container number in msg container

set shipment type in msg container

set TareWeight in msg container

set goods weight in msg container

set storage location in msg container

set remarks in msg container

set sealNo1 in msg container

set sealNo2 in msg container

set sealNo3 in msg container

case1.2# if get status of container is not exist:

add msg container to seasub

case1.3# if get status code of container is not exist or 'DR' or '':

add msg container to seasub

case1.4# if CustFinalDestCode of bill header is exist:

set port code and port type from (bill header)

if CustFinalDestCode of bill header is not exist:

set port code and port type (from shipment)

case1.5# if port code of billHeader is not exist:

set port id

case1.6# if submit type of manifest equal '0' and check header available is true:

set sequence number in routes

set country code in routes

set port code in routes

set port type in routes

set remarks in routes

add routes to seasub

case1.7# if submit type of manifest equal '0' and check header available is true:

set header in seasub

set record reference in seaMafRec

set Representative

set DocType to "MAF" in manifest

set MsgType to "SEASUB" in manifest

set MsgID

return meg

return manifest and message type

## Step #3.3: get generator

accepts massage and 1 as number

Case #1: if format equal one: get generator

That take massage type “seasub” and document type “Maf”

get path from "xml/XMLGenerators.properties"

case #1.2: if path start with "xml"

key ="SAU.XML.GEN."

className = key + documentType +'.'+massageType

make instance of generator using class name

## Step #3.4: update

- CASE #1: check if manifest header is available

**"update tsau\_maf\_status\_base set status = ? where acctid = ? and docrefno = ? and sno = ?"**

query parameters:

1 - maf status "ST" [staticly inserted]

2 - manifest account id [from xml message]

3 - manifest document reference number [from change [22] in database]

4 - sequence number "1"

get manifest routes list

**"update tsau\_maf\_route\_base set status = ? where acctid = ? and docrefno = ?"**

query parameters:

1 - maf status "ST" [staticly inserted]

2 - manifest account id [from xml message]

3 - manifest document reference number [from change[22] in database]

**"update tsau\_maf\_shipagt\_base set status = ? where acctid = ? and docrefno = ?"**

query parameters:

1 - maf status "ST" [staticly inserted]

2 - manifest account id [from xml message]

3 - manifest document reference number [from change[22] in database]

get manifest bills list

get manifest agent list

iterate over agent list, for every agent:

get agent list of bills, if list doesn't exist break from the loop

get manifest list of containers

**"select sno from tsau\_maf\_container\_base where acctid = ? and docrefno = ? and** **shipagentno = ?"**

query parameters:

1 - manifest account id [from xml message]

2 - manifest document reference number [from change[22] in database]

3 - agent key string [from xml message manifest agents]

**"update tsau\_maf\_container set status = ? where acctid = ? and docrefno = ? and sno = ?"**

query parameters:

1 - manifest status "ST" [staticly inserted]

2 - manifest account id [from xml message]

3 - manifest document reference number [from change[22] in database]

4 - unknown parameter

**"select t2.acctid, t2.docrefno from tsau\_maf\_details\_base t1, tsau\_bl\_header t2 where t1.bl\_docacctid= t2.acctid and t1.bl\_docrefno= t2.docrefno and t1.acctid= ? and t1.docrefno= ? and t2.shipagentno = ?"**

query parameters:

1 - manifest account id [from xml message]

2 - manifest document reference number [from change[22] in database]

3 - agent key string [from xml message manifest agent list]

**"update tsau\_bl\_header\_base set docstatus = ? where acctid = ? and docrefno = ?"**

query parameters:

1 - manifest status "ST" [staticly inserted]

2 - manifest document reference number [from change[22] in database]

3 - agent key string [from xml message manifest agent list]

## STEP #3.5: inserting in message history

Use this query to insert those values in table **TSAU\_DOC\_MSGHISTORY**

**"INSERT INTO TSAU\_DOC\_MSGHISTORY (ACCTID,DOCREFNO,MSGID,DOCTYPE,MSGTYPE,INOUTFLAG,HISTORYDATE,ORIGINAL\_MSG\_TYPE,STATUS,FILLER4,FILLER1,FILLER2)**

**values(?,?,?,?,?,?,?,?,?,?,?,?)"**

**Query parameters:**

**1 - header account id [from xml message]**

**2 - header document reference number [change [22] in database]**

**3 - message id [changed in TRANSFORMER STEP]**

**4 - document type “MAF” [changed in TRANSFORMER STEP]**

**5 - message type “SEASUB” [changed in TRANSFORMER STEP]**

**6 - history in and out “O”[staticly inserted]**

**7 – current time stamp [calculated]**

**8 - history original message type [unknown]**

**9 - history status [unknown]**

**10 - maf user info user id [changed in TRANSFORMER STEP]**

**11 - history filler1[unknown]**

**12 - history filler2 [unknown]**

## Setp 3.6 : submitAuditManifest

### Step #3.6.1 :

Set the type of IN\_OUT msg is O

Case 1:

- If msgID is not exist or the its length is less than 1

- If destination address is not exist or its length is less than 1

- If channel is not exist or its length is less than 1

All cases above will send assertion message ‘insufficient or Invalid Data in Msg Obj’

### Step #3.6.2 : write to file

#### Step #3.6.2.1: set msg file path

Case 1:

- If IN\_OUT msg is I

- the file path it will be path of Inbox address and the date path of the msg

Case 2:

- If IN\_OUT msg is O

- the file path it will be path of Outbox address the date path of the msg

Case 3:

- Else the file path it will be path of Misc address the date path of the msg

#### Step #3.6.2.2: set msg file path

Case 1:

- If msg ID is exist and its length is greater than 0

- The file name will be ‘msgID + . + fileType’

Case 2:

- Else if jms time stamp is exist and its length is greater than 0

- The file name will be ‘JmsTimeStamp + . + fileType’

Case 3:

- The file name will be ‘originAddress + ms + . + fileType’

Ms : is current time millis

### Step #3.6.3: send the msg

#### Step #3.6.3.1: The sender remote is &quot;ejb/sau/MsgSenderHome&quot;

#### Step #3.6.3.2: send the message to the sender remote

Case 1:

- If msg channel is exist and it is not WS

- Connect to the DB URL &quot;java:comp/env/jdbc/sauDSLocal&quot;

Case 1.1:

- Channel is not TC

##### Step# 3.6.3.2.1 :method : insert

accepts : message

##### step 3.6.3.2.2Get channel name about this message

Case1#

if channel is not null and not equal ‘WS’ : insert in database

INSERT INTO TSAU\_MSGES “MSG\_ID, IN\_OUT, INS\_TIME, PROC\_STATUS, MSG\_FILENAME, MSG\_FILEPATH, JMS\_TIMESTAMP, REF\_ID, STATUS\_C\_TIME, CHANNEL, MSG\_FORMAT, DOCTYPE, MSGTYPE, ORIGINAL\_MSG\_FILENAME, ORIGIN\_ADDRESS, DEST\_ADDRESS” VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)

Query Parameters :

1 - MSG\_ID: from xml message

2 - IN\_OUT: fixed value ‘o’;

3 - INS\_TIME: get current timestamp (calculated)

4 - PROC\_STATUS: fixed value ‘N’;

5 - MSG\_FILENAME: get from message

6 - MSG\_FILEPATH: get from message

7 - JMS\_TIMESTAMP: from xml message

8 - REF\_ID: from xml message

9 - STATUS\_C\_TIME: from xml message

10 - CHANNEL: get from msgtransformer ‘MQ’

11 - MSG\_FORMAT: from xml message

12 - DOCTYPE: get from msgtransformer ‘MAF’

13 - MSGTYPE: get from msgtransformer ‘SEASUB’

14 - ORIGINAL\_MSG\_FILENAME: get from msgtransformer

15 – ORIGIN\_ADDRESS: get from from msgtransformer

16 – DEST\_ADDRESS: get from msgtransformer

Case 1.2: if msg channel is MQ

- Create the queue sender and set the portJndi by using the msg destination

address

- If destination address in not exist, we will use the msg address

- Send the msg EDIMsg

### Step #3.6.4: update the status of MsgDAO

“ UPDATE TSAU\_MSGES SET PROC\_STATUS=?, STATUS\_C\_TIME=? WHERE MSG\_ID=? “

Query parameters :

1. Status “F” [statically inserted]

2. Time stamp current time [calculated]

3. Message id []

### The step#3.6.5 will returned Prepared Statement of the connection as result to the step#1

### Step #3.6.6 : Submit audit manifest

- Get locale English and locale Arabic

- Get Subject English and Arabic formatted Message and Act English and Arabic

formatted Message

Create an audit info and set the values :

- DocRefNo : from manifest header

- Sau\_Accid: from manifest header

- Module “9”: [statically inserted]

- Activity : [get from Act English]

- Activity\_ar : [get from Act Arabic]

- Remarks: [get from locale English]

- Remarks\_ar: [get from locale Arabic]

- ActivityDoneBy: from manifest header

- ActivityDate: current date

- Port: from manifest header

- Sau\_UserId: from manifest header