|  |
| --- |
| Smartstream Technologies |
| Excalibur BatchPedia |
| Technical Guide |

|  |
| --- |
| Asokan Srinivasan  7/29/2016 |

Contents

[Cancel and Correct Batch: 7](#_Toc459125012)

[Pre-requisite 7](#_Toc459125013)

[How to run: 7](#_Toc459125014)

[Post Check 8](#_Toc459125015)

[Debug 8](#_Toc459125016)

[Clean up 8](#_Toc459125017)

[Scenario 1 8](#_Toc459125018)

[Scenario 2 8](#_Toc459125019)

[Configurations / Points to Note 8](#_Toc459125020)

[Enrichment Batch: 9](#_Toc459125021)

[Pre-requisite 9](#_Toc459125022)

[How to run: 9](#_Toc459125023)

[Post Check 9](#_Toc459125024)

[Debug 10](#_Toc459125025)

[Clean up 10](#_Toc459125026)

[Scenario 1 10](#_Toc459125027)

[Scenario 2 10](#_Toc459125028)

[Configurations / Points to Note 10](#_Toc459125029)

[Calculation Batch: 10](#_Toc459125030)

[Pre-requisite 10](#_Toc459125031)

[How to run: 11](#_Toc459125032)

[Calc 11](#_Toc459125033)

[Post Check 11](#_Toc459125034)

[Debug 12](#_Toc459125035)

[Recalc / Clean up 12](#_Toc459125036)

[Scenario 1 13](#_Toc459125037)

[Scenario 2 13](#_Toc459125038)

[Scenario 3 13](#_Toc459125039)

[Configurations / Points to Note 14](#_Toc459125040)

[Transaction Charge Staging Batch: 17](#_Toc459125041)

[Pre-requisite 17](#_Toc459125042)

[How to run: 18](#_Toc459125043)

[Post Check 19](#_Toc459125044)

[Debug 19](#_Toc459125045)

[Clean up 19](#_Toc459125046)

[Scenario 1 19](#_Toc459125047)

[Scenario 2 20](#_Toc459125048)

[Configurations / Points to Note 20](#_Toc459125049)

[Transaction Journal / Trade journal Batch: 20](#_Toc459125050)

[Pre-requisite 20](#_Toc459125051)

[How to run: 21](#_Toc459125052)

[Post Check 21](#_Toc459125053)

[Debug 22](#_Toc459125054)

[Clean up 22](#_Toc459125055)

[Scenario 1 22](#_Toc459125056)

[Scenario 2 22](#_Toc459125057)

[Configurations / Points to Note 22](#_Toc459125058)

[Manual Accrual to Journal Summary Batch: 23](#_Toc459125059)

[Pre-requisite 23](#_Toc459125060)

[How to run: 23](#_Toc459125061)

[Post Check 23](#_Toc459125062)

[Debug 23](#_Toc459125063)

[Clean up 23](#_Toc459125064)

[Scenario 1 23](#_Toc459125065)

[Scenario 2 23](#_Toc459125066)

[Configurations / Points to Note 23](#_Toc459125067)

[Transaction Journal to Journal Summary Batch: 24](#_Toc459125068)

[Pre-requisite 24](#_Toc459125069)

[How to run: 24](#_Toc459125070)

[Post Check 24](#_Toc459125071)

[Debug 24](#_Toc459125072)

[Clean up 24](#_Toc459125073)

[Scenario 1 24](#_Toc459125074)

[Scenario 2 24](#_Toc459125075)

[Configurations / Points to Note 24](#_Toc459125076)

[Trade Journal to Journal Summary Batch: 25](#_Toc459125077)

[Pre-requisite 25](#_Toc459125078)

[How to run: 25](#_Toc459125079)

[Post Check 25](#_Toc459125080)

[Debug 25](#_Toc459125081)

[Clean up 25](#_Toc459125082)

[Scenario 1 25](#_Toc459125083)

[Scenario 2 25](#_Toc459125084)

[Configurations / Points to Note 25](#_Toc459125085)

[Expense Summary Batch: 26](#_Toc459125086)

[Pre-requisite 26](#_Toc459125087)

[How to run: 26](#_Toc459125088)

[Post Check 26](#_Toc459125089)

[Debug 28](#_Toc459125090)

[Clean up 28](#_Toc459125091)

[Scenario 1 28](#_Toc459125092)

[Scenario 2 28](#_Toc459125093)

[Configurations / Points to Note 28](#_Toc459125094)

[GL Enrichment Batch: 29](#_Toc459125095)

[Pre-requisite 29](#_Toc459125096)

[How to run: 29](#_Toc459125097)

[Post Check 29](#_Toc459125098)

[Debug 29](#_Toc459125099)

[Clean up 30](#_Toc459125100)

[Scenario 1 30](#_Toc459125101)

[Scenario 2 30](#_Toc459125102)

[Configurations / Points to Note 30](#_Toc459125103)

[Allocation Batch: 31](#_Toc459125104)

[Pre-requisite 31](#_Toc459125105)

[How to run: 31](#_Toc459125106)

[Post Check 32](#_Toc459125107)

[Debug 32](#_Toc459125108)

[Clean up 32](#_Toc459125109)

[Scenario 1 32](#_Toc459125110)

[Scenario 2 32](#_Toc459125111)

[Configurations / Points to Note 32](#_Toc459125112)

[Postings Batch: 33](#_Toc459125113)

[Pre-requisite 33](#_Toc459125114)

[How to run: 33](#_Toc459125115)

[Post Check 33](#_Toc459125116)

[Debug 33](#_Toc459125117)

[Clean up 33](#_Toc459125118)

[Scenario 1 : 33](#_Toc459125119)

[Creating ELC Approved Records and Running Posting Batch for Period and Product without cleaning of Already Posted Records 33](#_Toc459125120)

[Scenario 2 33](#_Toc459125121)

[Creating ELC Approved Records and Running Posting Batch for Peroid and Product cleaning of Already Posted Records 33](#_Toc459125122)

[Configurations / Points to Note 33](#_Toc459125123)

[Trade Line Detail Batch: 34](#_Toc459125124)

[Pre-requisite 34](#_Toc459125125)

[How to run: 34](#_Toc459125126)

[Post Check 34](#_Toc459125127)

[Debug 34](#_Toc459125128)

[Clean up 35](#_Toc459125129)

[Scenario 1 35](#_Toc459125130)

[Scenario 2 35](#_Toc459125131)

[Scenario 3 35](#_Toc459125132)

[Configurations / Points to Note 35](#_Toc459125133)

[Invoice Accrual Line Detail Batch: 36](#_Toc459125134)

[Pre-requisite 36](#_Toc459125135)

[How to run: 36](#_Toc459125136)

[Post Check 36](#_Toc459125137)

[Debug 36](#_Toc459125138)

[Clean up 36](#_Toc459125139)

[Scenario 1 : 36](#_Toc459125140)

[Scenario 2 37](#_Toc459125141)

[Scenario 2 37](#_Toc459125142)

[Configurations / Points to Note 37](#_Toc459125143)

[Invoice Trade Auto-match Batch: 38](#_Toc459125144)

[Pre-requisite 38](#_Toc459125145)

[How to run: 38](#_Toc459125146)

[Method 1 38](#_Toc459125147)

[Method 2 38](#_Toc459125148)

[Post Check 39](#_Toc459125149)

[Method 1 39](#_Toc459125150)

[Method 2 39](#_Toc459125151)

[Debug 39](#_Toc459125152)

[Clean up 39](#_Toc459125153)

[Scenario 1 39](#_Toc459125154)

[Scenario 2 40](#_Toc459125155)

[Solution 40](#_Toc459125156)

[Configurations / Points to Note 40](#_Toc459125157)

[Invoice Summary Auto-Match Batch: 41](#_Toc459125158)

[Pre-requisite 41](#_Toc459125159)

[How to run: 41](#_Toc459125160)

[Post Check 42](#_Toc459125161)

[Method 1 42](#_Toc459125162)

[Method 2 42](#_Toc459125163)

[Debug 42](#_Toc459125164)

[Clean up 42](#_Toc459125165)

[Scenario 1 42](#_Toc459125166)

[Scenario 2 43](#_Toc459125167)

[Solution 43](#_Toc459125168)

[Configurations / Points to Note 43](#_Toc459125169)

[Charge Clearance Queries By Party 44](#_Toc459125170)

[Accrual Clearance Queries By Party 45](#_Toc459125171)

[Find the count of data and note down 45](#_Toc459125172)

[For Trade Based Accruals 46](#_Toc459125173)

[Checking Count 46](#_Toc459125174)

[Cleanup Queries 46](#_Toc459125175)

[Invoice Clearance Queries 46](#_Toc459125176)

[Clean Up Queries 47](#_Toc459125177)

[Trade Charges 47](#_Toc459125178)

[Search Query for CRDRV 47](#_Toc459125179)

[Search Query for FX 47](#_Toc459125180)

# Cancel and Correct Batch:

## Pre-requisite

To check records available of CnC for Product and Period

*SELECT COUNT(\*),*

*PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS*

*FROM EXAMIN\_TRANSACTION ET,*

*TRANSACTION\_REFERENCE TR*

*WHERE ET.TRANS\_ID = TR.TRANS\_ID*

*AND ET.TRANS\_VERSION = TR.TRANS\_VERSION*

*AND ET.PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND ET.TRADE\_DATE BETWEEN '01-DEC-2015' AND '31-DEC-2015'*

*and PROCESS\_STATE\_CD is null*

*and ET.CALC\_ELIGIBLE\_IND = 'N';*

Generic Query

*SELECT COUNT(\*),*

*PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS*

*FROM EXAMIN\_TRANSACTION ET,*

*TRANSACTION\_REFERENCE TR*

*WHERE ET.TRANS\_ID = TR.TRANS\_ID*

*AND ET.TRANS\_VERSION = TR.TRANS\_VERSION*

*AND ET.PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND ET.TRADE\_DATE BETWEEN '01-DEC-2015' AND '31-DEC-2015'*

*group by PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS;*

## How to run:

export SCRIPT\_NAME=CNC

./startNewBatch.sh -DbatchName=CncDistributor -DbatchType=Daily -DproductTypes=FX -DstartDate=12/01/2015

To run for entire month

./startNewBatch.sh -DbatchName=CncDistributor -DbatchType=Daily -DproductTypes=FX -DstartDate=12/01/2015 -DendDate=12/31/2015

The batch updates the business date in examin\_transaction to the “currentDate” in Examin\_Business\_Date

## Post Check

*SELECT COUNT(\*),*

*PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS*

*FROM EXAMIN\_TRANSACTION ET,*

*TRANSACTION\_REFERENCE TR*

*WHERE ET.TRANS\_ID = TR.TRANS\_ID*

*AND ET.TRANS\_VERSION = TR.TRANS\_VERSION*

*AND ET.PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND ET.TRADE\_DATE BETWEEN '01-DEC-2015' AND '31-DEC-2015'*

*and PROCESS\_STATE\_CD is '*CnCed*'*

*and ET.CALC\_ELIGIBLE\_IND = 'Y';*

## Debug

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Status | PROCESS\_STATE\_CD | TRANS\_ACTIVE\_IND | CALC\_ELIGIBLE\_IND | CALC\_IND | TRANS\_EXCEPTION\_IND | TRANS\_CHARGE\_STATUS |
| Fresh Record & Ready for Cancel N Correct | NULL/BLANK | Y | N | N | N | NULL/BLANK |
| After Cancel N Correct & Ready for Enrichment | CnCed | Y | Y | N | N | NULL/BLANK |

Following is the different record status in Transaction\_Reference table before and after the CnC

Between EQ and NON-EQ, the difference is Cancel N Correct and Enrichment Jobs are not run for EQ. Rest of the batch remains the same. For EQ the Daily calc simply skips CncDistributor and Enrichment. Rest of the activities remains the same that is , Running Total Initilization Pre and Post, Calculation and Updation of rows.

# Enrichment Batch:

## Pre-requisite

To check records available of CnC for Product and Period

*SELECT COUNT(\*),*

*PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS*

*FROM EXAMIN\_TRANSACTION ET,*

*TRANSACTION\_REFERENCE TR*

*WHERE ET.TRANS\_ID = TR.TRANS\_ID*

*AND ET.TRANS\_VERSION = TR.TRANS\_VERSION*

*AND ET.PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND ET.TRADE\_DATE BETWEEN '01-DEC-2015' AND '31-DEC-2015'*

*and PROCESS\_STATE\_CD is '*CnCed*'*

*and ET.CALC\_ELIGIBLE\_IND = 'Y';*

## How to run:

export SCRIPT\_NAME=ENRICHMENT

./startNewBatch.sh -DbatchName=EnrichEngineMain -DbatchType=Daily -DproductTypes=FX -DstartDate=12/01/2015

## Post Check

*SELECT COUNT(\*),*

*PROCESS\_STATE\_CD,*

*ET.TRANS\_ACTIVE\_IND,*

*ET.CALC\_ELIGIBLE\_IND,*

*TR.TRANS\_EXCEPTION\_IND,*

*ET.TRANS\_CHARGE\_STATUS*

*FROM EXAMIN\_TRANSACTION ET,*

*TRANSACTION\_REFERENCE TR*

*WHERE ET.TRANS\_ID = TR.TRANS\_ID*

*AND ET.TRANS\_VERSION = TR.TRANS\_VERSION*

*AND ET.PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND ET.TRADE\_DATE BETWEEN '01-DEC-2015' AND '31-DEC-2015'*

*and PROCESS\_STATE\_CD is 'Enriched'*

*and ET.CALC\_ELIGIBLE\_IND = 'Y';*

## Debug

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

Following is the different record status in Transaction\_Reference table before and after the Enrichment Batch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Status | PROCESS\_STATE\_CD | TRANS\_ACTIVE\_IND | CALC\_ELIGIBLE\_IND | CALC\_IND | TRANS\_EXCEPTION\_IND | TRANS\_CHARGE\_STATUS |
| After Cancel N Correct & Ready for Enrichment | CnCed | Y | Y | N | N | NULL/BLANK |
| After Enrichment & Ready for Calculation | Enriched | Y | Y | N | N | NULL/BLANK |

# Calculation Batch:

## Pre-requisite

*SELECT COUNT(1)*

*FROM EXAMIN\_TRANSACTION et,*

*TRANSACTION\_REFERENCE tr*

*WHERE (et.BUSINESS\_DATE >='05-Aug-2015' )*

*AND(et.BUSINESS\_DATE <='05-Aug-2015' )*

*AND(et.TRANS\_ACTIVE\_IND ='Y' )*

*AND(et.CALC\_ELIGIBLE\_IND ='Y' )*

*AND(et.CALC\_IND ='N' )*

*AND(tr.TRANS\_EXCEPTION\_IND ='N'*

*AND et.TRANS\_ID =tr.TRANS\_ID*

*AND et.TRANS\_VERSION =tr.TRANS\_VERSION)*

*AND(et.PROCESS\_STATE\_CD IN('Enriched' , 'Calced'))*

*AND(et.PRODUCT\_TYPE\_INTRNL\_ID IN(?)) ;*

## How to run:

### Calc

calc.sh [-r] [-s <start\_date>] [[-e <end\_date>] | [-d <duration>]] (FX|LEO|CRDRV|EQDRV|USEQ|EUEQ|BANKCHARGE|APEQ)

./calc.sh -s 07/29/2015 -e 08/05/2015 -d -0 FUTURES

Manually executing batch

./calc.sh -r -s 12/01/2015 -d -0 FX

In case Running\_total\_initilization to be run Manually

export SCRIPT\_NAME=PreRunnTotalInit

./startNewBatch.sh -DbatchName=RunningTotalInitializer -DproductTypes=FX -DbatchType=Daily -DstartDate=12/01/2015 -DpreOrPostRun=pre -Dduration=0 -DrunningTotalIds=568444

## Post Check

TRANSACTION\_CHARGE

*Select count(\*) from transaction\_charge where product\_type\_intrnl\_id=?*

*and party\_intrnl\_id=?*

*and charge\_type\_intrnl\_id=?*

*and primary\_calc\_rule\_id=?*

TRANSACTION\_CHARGE\_RATE

*SELECT \**

*FROM TRANSACTION\_CHARGE\_RATE*

*WHERE (TRANS\_CHARGE\_ID, TRANS\_CHARGE\_VERSION) IN*

*(SELECT TRANS\_CHARGE\_ID,*

*TRANS\_CHARGE\_VERSION*

*FROM TRANSACTION\_CHARGE*

*WHERE PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND TRADE\_DATE BETWEEEN '01-Jan-2016'*

*AND '01-Jan-2016'*

*);*

TRANSACTION\_LIMIT\_RATE

*SELECT \**

*FROM TRANSACTION\_LIMIT\_RATE*

*WHERE (TRANS\_CHARGE\_ID, TRANS\_CHARGE\_VERSION) IN*

*(SELECT TRANS\_CHARGE\_ID,*

*TRANS\_CHARGE\_VERSION*

*FROM TRANSACTION\_CHARGE*

*WHERE PRODUCT\_TYPE\_INTRNL\_ID IN (101,102,103)*

*AND TRADE\_DATE BETWEEEN '01-Jan-2016'*

*AND '01-Jan-2016'*

*);*

LIMIT\_TIER\_RUNNING\_TOTAL

*SELECT \**

*FROM LIMIT\_TIER\_RUNNING\_TOTAL*

*WHERE RUNNING\_TOTAL\_ID IN*

*(SELECT RUNNING\_TOTAL\_ID FROM RUNNING\_TOTAL\_RULE WHERE PARTY\_INTRNL\_ID = ?*

*);*

CHARGE\_RUNNING\_TOTAL

*SELECT \**

*FROM LIMIT\_TIER\_RUNNING\_TOTAL*

*WHERE RUNNING\_TOTAL\_ID IN*

*(SELECT RUNNING\_TOTAL\_ID FROM RUNNING\_TOTAL\_RULE WHERE PARTY\_INTRNL\_ID = ?*

*);*

EXCEPTION TABLES:

EXAMIN\_EXCEPTION, EXCEPTION\_DETAILS

*SELECT \**

*FROM EXAMIN\_EXCEPTION EE,*

*EXCEPTION\_DETAILS ED*

*WHERE EE.EXCEPTION\_ID = ED.EXCEPTION\_ID*

*AND EE.EXCEPTION\_TYPE ='Calculation'*

*AND PRODUCT\_TYPE\_INTRNL\_ID (101,102,103)*

*AND EE.STATUS='New';*

## Debug

1. Check with a Transaction id in Calc Debug screen
2. If no charges to be calculated, then Charge Rules to be set
3. If a Calc Rule is not found for specific charges, set up Calculation Rule
4. Check the logs for any exceptions
5. Check the Exceptions 🡪 Calculation Tab for any transactions in exception state

## Recalc / Clean up

If a Calc has to be re-run

### Scenario 1

New Transaction version is created

#### Solution

1. **Do not** clean the charges.
2. Run the Calc Job as in the How to Run section. New Transaction Charge Versions should be created

### Scenario 2

Rate Schedules have undergone changes – User did not request recalc when prompted

#### Solution

To Write about : Groovy run with Calc Rule

### Scenario 3

Rate Schedules have undergone changes – User have request recalc when prompted Running Totals are not right. One possible reason, they were not initialised during previous calc

#### Solution

Recalc is calculated on daily basis. This happens in two steps, First step is a setup and second setup in Recalc.

Input tables:

RATE\_RECALC\_REQUEST

RECALC\_REQUEST

Output tables:

TRANSACTION\_CHARGE

TRANSACTION\_CHARGE\_RATE

TRANSACTION\_LIMIT\_RATE

LIMIT\_TIER\_RUNNING\_TOTAL

CHARGE\_RUNNING\_TOTAL

--Step 1

Daily ReCalc setup

-DbatchName=RateRecalcDistributor -DbatchType=Daily

--Step 2

Daily Recalc

export SCRIPT\_NAME=RecalcRequestFX

./startNewBatch.sh -DbatchName=ReCalcRequestEngine -DbatchType=Daily -DproductTypes=FX -DrunCleanup=yes

Other possible combinations for Daily Recalc

-DproductTypes=CRDRV

-DproductTypes=EQDRV

-DproductTypes=LEO

-DproductTypes=EQ

-DproductTypes=IRDRV

-DproductTypes=BANKCHARGE

export SCRIPT\_NAME=RecalcUpdateTransStatus

./startNewBatch.sh -DbatchName=RecalcUpdateTransactionStatus -DbatchType=Daily -DproductTypes=EQ,IRDRV,FX,CRDRV,EQDRV

## Configurations / Points to Note

Following is the different record status in Transaction\_Reference table before and after the Enrichment Batch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Status | PROCESS\_STATE\_CD | TRANS\_ACTIVE\_IND | CALC\_ELIGIBLE\_IND | CALC\_IND | TRANS\_EXCEPTION\_IND | TRANS\_CHARGE\_STATUS |
| After Enrichment & Ready for Calculation | Enriched | Y | Y | N | N | NULL/BLANK |
| After Calculation | Calced | Y | Y | Y | N | Charges |
| Calculation-Exception | Calced | Y | Y | N | Y | NULL/BLANK |

Calculation batch is for calculating Charges for each Transaction/Trades.

Input for calculation batch is

* EXAMIN\_TRANSACTION
* TRANSACTION\_REFERENCE,
* SATELLITE TABLES (EQ\_CASH\_TRANSACTION for EQCASH, FX\_CASH\_TRANSACTION for FXCASH etc…)

Output to

* TRANSACTION\_CHARGE
* TRANSACTION\_CHARGE\_RATE
* TRANSACTION\_LIMIT\_RATE
* LIMIT\_TIER\_RUNNING\_TOTAL
* CHARGE\_RUNNING\_TOTAL

Rule Tables

* CALCULATION\_RULE
* CALCULATION\_RULE\_BASIS
* RATE\_SCHEDULE
* RUNNING\_TOTAL\_RULE
* RUNNING\_TOTAL\_BASIS
* LIMIT\_RULE
* LIMIT\_RULE\_BASIS
* LIMIT\_SCHEDULE

Calculation Batch are broadly categorised under two titles

* EQ
* Non-EQ batchs.

Following are the universe of batchs

* Cancel And Correct
* Enrichment
* RunningTotalInitilizer-Pre
* Calculation
* RunningTotalInitilizer-Post
* UpdateTransactions

|  |  |  |
| --- | --- | --- |
| EQ – Calculation | Non-EQ Calculation | Where clauses |
|  | com.cs.examin.batch.calcengine.CncDistributor | et.businessDate >= :startDate and et.businessDate <= :endDate and et.transActiveInd = 'Y' and et.processStateCd is null |
|  | com.cs.examin.batch.enrichmentengine.EnrichmentEngine | et.businessDate>=:startDate and et.businessDate<=:endDate and et.processStateCd='CnCed' and et.calcEligibleInd='Y' and et.transActiveInd='Y' |
| RunningTotalInitilizer-Pre | RunningTotalInitilizer-Pre | com.cs.examin.batch.runningtotal.RunningTotalInitializer |
| Calculation | com.cs.examin.batch.calcengine.CalcEngineDistributor | et.businessDate >=:startDate and et.businessDate <=:endDate and et.transActiveInd = 'Y' and et.calcEligibleInd='Y' and et.calcInd='N' and et.processStateCd in ('Enriched', 'Calced') and et.transactionReference.transExceptionInd ='N' |
| RunningTotalInitilizer-Post | RunningTotalInitilizer-Post | com.cs.examin.batch.runningtotal.RunningTotalInitializer |
|  | com.cs.examin.batch.calcengine.UpdateTransactionStatusDistributor | et.businessDate>=:startDate AND et.businessDate <=:endDate AND et.calcEligibleInd='Y' AND et.calcInd='N' AND et.transActiveInd = 'Y' AND et.id.transId = tr.id.transId AND et.id.transVersion = tr.id.transVersion AND tr.transExceptionInd != 'Y' |

Commands to execute a Daily batch called from Contrl+M

./dailyUSEQCalc.sh

./calc.sh USEQ

./dailyNonEQCalc.sh

./ dailyCnC.sh

./ enrichData.sh

./ calc.sh FX CRDRV EQDRV IRDRV

# Transaction Charge Staging Batch:

## Pre-requisite

1. Accrual Eligibility Rule
   1. Rule should be set up for particular productType and productSubType.
   2. The effective date should be grteater than the posting period or Startdate
   3. No. of posting business days should be equal to the NUM\_BUSINESS\_DAYS in examin\_business\_date table. Use the below query to validate: select \* from examin\_business\_date where DATE\_TYPE='CurrentDate';
   4. Rule Criteria : The transactions should pass the rule criteria conditions.
2. If running the Recalc request , check the charge\_calc\_date in the Transaction\_charge\_staging table for the existing record.

**Query To Get Accrual Eligibility Rules :**

1. **BatchType: Monthly**

*SELECT ar.ACCRUAL\_RULE\_ID ,*

*ar.ACCRUAL\_RULE\_VERSION ,*

*ae.ACCRUAL\_ELIGIBILITY\_ID ,*

*ae.ACCRUAL\_ELIGIBILITY\_VERSION ,*

*FROM ACCRUAL\_RULE ar,*

*ACCRUAL\_ELIGIBILITY ae,*

*PRODUCT\_TYPE pt,*

*CHARGE\_TYPE ct,*

*EXAMIN\_BUSINESS\_DATE ebd*

*WHERE ar.ACCRUAL\_RULE\_ID =ae.ACCRUAL\_RULE\_ID*

*AND ar.RECORD\_ACTIVE\_IND ='Y'*

*AND ae.RECORD\_ACTIVE\_IND ='Y'*

*AND ae.ELIGIBLE\_IND ='Y'*

*AND ar.PRODUCT\_TYPE\_INTRNL\_ID=pt.PRODUCT\_TYPE\_INTRNL\_ID*

*AND ar.CHARGE\_TYPE\_INTRNL\_ID =ct.CHARGE\_TYPE\_INTRNL\_ID*

*AND ebd.DATE\_TYPE ='CurrentDate'*

*AND ebd.Num\_Business\_Days =ae.POSTING\_BUSINESS\_DAY*

*AND NOT (EXISTS*

*(SELECT 1*

*FROM EXAMIN\_STATIC\_REFERENCE esr*

*WHERE esr.TOKEN\_TYPE ='AccrualRule'*

*AND esr.TOKEN\_SUB\_TYPE ='ETLOnly'*

*AND esr.PRODUCT\_TYPE\_CODE =pt.PRODUCT\_TYPE\_CODE*

*AND esr.PRODUCT\_SUB\_TYPE\_CODE=pt.PRODUCT\_SUB\_TYPE\_CODE*

*AND esr.CHARGE\_TYPE\_CODE =ct.CHARGE\_TYPE\_CODE*

*AND esr.CHARGE\_SUB\_TYPE\_CODE =ct.CHARGE\_SUB\_TYPE\_CODE*

*))*

*AND (ar.PRODUCT\_TYPE\_INTRNL\_ID IN (134 , 133 , 131 , 132))*

*AND (ar.EFF\_DATE IS NULL*

*OR ar.EFF\_DATE <=? )*

*AND NVL(ar.EXP\_DATE, '01-Jan-2999')>=?*

1. **BatcType : Daily**

*SELECT ar.ACCRUAL\_RULE\_ID,*

*ar.ACCRUAL\_RULE\_VERSION ae.ACCRUAL\_ELIGIBILITY\_ID,*

*ae.ACCRUAL\_ELIGIBILITY\_VERSION ,*

*FROM ACCRUAL\_RULE ar,*

*ACCRUAL\_ELIGIBILITY ae*

*WHERE ar.ACCRUAL\_RULE\_ID =ae.ACCRUAL\_RULE\_ID*

*AND ar.RECORD\_ACTIVE\_IND ='Y'*

*AND ae.RECORD\_ACTIVE\_IND ='Y'*

*AND ae.ELIGIBLE\_IND ='Y'*

*AND NVL(ae.NO\_OF\_PREV\_MONTHS, 0) >0*

*AND ( ar.PRODUCT\_TYPE\_INTRNL\_ID IN (134 , 133 , 131 , 132) )*

*AND ( ar.EFF\_DATE IS NULL*

*OR ar.EFF\_DATE <=? )*

*AND NVL(ar.EXP\_DATE, '01-Jan-2999')>=?*

1. **Query To Check the Eligible records :**

*SELECT count(\*) FROM EXAMIN\_TRANSACTION ET, TRANSACTION\_CHARGE TC, IR\_DERIVATIVE\_TRANSACTION IR , PARTY\_REFERENCE PR*

*WHERE TC.PARTY\_INTRNL\_ID = PR.PARTY\_INTRNL\_ID*

*AND ET.TRANS\_ID = TC.TRANS\_ID*

*AND ET.TRANS\_VERSION = TC.TRANS\_VERSION*

*AND IR.TRANS\_ID = ET.TRANS\_ID*

*AND IR.TRANS\_VERSION = ET.TRANS\_VERSION*

*AND TC.LEGAL\_ENTITY\_INTRNL\_ID IN ( 88153 , 88669 , 88394 ) --- LegalEntity Id from Accral Eligibility rule criteria*

*AND TC.PARTY\_INTRNL\_ID IN ( 43182 , 21053369 , 1517778 ) --- PartyIntrnlId from Accral Eligibility rule criteria*

*AND PR.RECORD\_ACTIVE\_IND ='Y' AND TC.PRODUCT\_TYPE\_INTRNL\_ID =133 –-- ProductType Intrnl ID*

*AND TC.CHARGE\_TYPE\_INTRNL\_ID =201 --- Charge type from accrual eligibility rule*

*AND IR.TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014' --- Trade date from the parameners*

*AND (( IR.TRADE\_DATE >= ?) and AND (( IR.TRADE\_DATE =< ?) --- Effective and end date of the rule*

*OR IR.TRADE\_DATE IS NULL ) AND TC.STAGING\_STATE\_CD IS NULL ;*

## How to run:

1. ./startNewBatch.sh -DbatchName=JournalAdjustmentStagingDistributor -DbatchType=Monthly -DproductTypes=IRDRV -DstartDate=01/01/2014 -DendDate=01/31/2014

-----The trade date values between the startdate and enddate

1. ./startNewBatch.sh -DbatchName=JournalAdjustmentStagingDistributor -DbatchType=Monthly -DproductTypes=IRDRV

---The trade date values between the First And Last date of Previous month from examin\_business\_date table.

1. ./startNewBatch.sh -DbatchName=JournalAdjustmentStagingDistributor -DbatchType=Monthly -DproductTypes=IRDRV -DpostingPeriod=01-2014

The trade date Values between The First and Last date of the posting period Month.

## Post Check

*SELECT COUNT(\*)*

*FROM transaction\_charge tc*

*WHERE (TC.TRANS\_CHARGE\_ID,TC.TRANS\_CHARGE\_VERSION) IN*

*(SELECT TRANS\_CHARGE\_ID,*

*TRANS\_CHARGE\_VERSION*

*FROM TRANSACTION\_CHARGE\_STAGING*

*WHERE RECORD\_STATUS IS NULL*

*AND ACCRUAL\_ELIGIBILITY\_ID =?*

*AND ACCRUAL\_ELIGIBILITY\_VERSION=?*

*);*

The count should match with precheck queries.

## Debug

1. Check the Accrual Eligibility Rule Effective date and End date in Schecule as well
2. Check the Examin Business Date
3. Check the No of Business days in Examin\_business\_date table and Posting Business days in Accrual Eligibility Rule.
4. Compare the Accrual Eligibility Rule Criteria and Column Values in Transaction Charge table.

## Clean up

*DELETE*

*FROM Transaction\_Charge\_staging*

*WHERE TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014'*

*AND PARTY\_INTRNL\_ID IN(?)*

*AND product\_type\_intrnl\_id IN(?);*

*UPDATE transaction\_charge TC*

*SET STAGING\_STATE\_CD = NULL*

*WHERE PARTY\_INTRNL\_ID IN ( 43182 )*

*AND TC.RECORD\_ACTIVE\_IND ='Y'*

*AND TC.PRODUCT\_TYPE\_INTRNL\_ID =131*

*AND TC.CHARGE\_TYPE\_INTRNL\_ID =201*

*AND TC.TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014';*

## Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

1. Check the below configuration the Examin\_static\_reference

Select \* from Examin\_static\_reference where token\_type=’BATCH’ and token\_sub\_type=’TransactionChargeStaging’ and product\_type\_code=?;

1. In the Transaction\_Charge table staging\_state\_cd should be null before running the batch.
2. After Batch runs the staging\_state\_cd will become ‘Staging’.
3. There is one Constrant in Transaction\_Charge\_Staging table, if record exists for Trans\_id, Trans\_Version, Charge\_calc\_date then it will throw the constraint violation exception.

**Date Logic :**

**BatchType : Daily**

1. **StartDate logic**

Checks for arguments "noOfMonthBack"

If noOfMonthBack is null it checks for Configuration : TokenType="JournalAdjustmentDistributor", token\_sub\_type='NoOfMonthBack' and gets no of Months back value.

* 1. If noOfMonthBack is not null then It takes the StartDate as businessDate with the negation of noOfMonthBack. Example : If BusinessDate=12-Dec-2015 and noOfMonthBack=4 Then the StartDate wil be 12-Aug-2015
  2. If "noOfMonthBack" is not defined it takes the start date as business Date

**2. EndDate logic**

Always the Business date as end date.

**BatchType : Monthly**

# Transaction Journal / Trade journal Batch:

## Pre-requisite

**Note : Date Logic is same as TransactionChargeStaging Batch**

1. Get the Accrual Eligibility Rules :
   1. **BatchType : Monthly**

*select ar.ACCRUAL\_RULE\_ID , ar.ACCRUAL\_RULE\_VERSION , ae.ACCRUAL\_ELIGIBILITY\_ID , ae.ACCRUAL\_ELIGIBILITY\_VERSION*

*from ACCRUAL\_RULE ar, ACCRUAL\_ELIGIBILITY ae, PRODUCT\_TYPE pt, CHARGE\_TYPE ct,EXAMIN\_BUSINESS\_DATE ebd*

*where ar.ACCRUAL\_RULE\_ID=ae.ACCRUAL\_RULE\_ID*

*and ar.RECORD\_ACTIVE\_IND='Y'*

*and ae.RECORD\_ACTIVE\_IND='Y'*

*and ae.ELIGIBLE\_IND='Y'*

*and ar.PRODUCT\_TYPE\_INTRNL\_ID=pt.PRODUCT\_TYPE\_INTRNL\_ID*

*and ar.CHARGE\_TYPE\_INTRNL\_ID=ct.CHARGE\_TYPE\_INTRNL\_ID*

*and ebd.DATE\_TYPE='CurrentDate'*

*and ebd.Num\_Business\_Days=ae.POSTING\_BUSINESS\_DAY*

*and not exists (select 1 from EXAMIN\_STATIC\_REFERENCE esr where esr.TOKEN\_TYPE='AccrualRule'*

*and esr.TOKEN\_SUB\_TYPE='ETLOnly' and esr.PRODUCT\_TYPE\_CODE=pt.PRODUCT\_TYPE\_CODE*

*and esr.PRODUCT\_SUB\_TYPE\_CODE=pt.PRODUCT\_SUB\_TYPE\_CODE and esr.CHARGE\_TYPE\_CODE=ct.CHARGE\_TYPE\_CODE*

*and esr.CHARGE\_SUB\_TYPE\_CODE=ct.CHARGE\_SUB\_TYPE\_CODE) and (ar.PRODUCT\_TYPE\_INTRNL\_ID in (134 , 133 , 131 , 132))*

*and (ar.EFF\_DATE is null or ar.EFF\_DATE<='01-Jan-2001'*

*) and nvl(ar.EXP\_DATE, '01-Jan-2999')>='01-Jan-2021';BathType : Daily*

*select ar.ACCRUAL\_RULE\_ID, ar.ACCRUAL\_RULE\_VERSION ae.ACCRUAL\_ELIGIBILITY\_ID,ae.ACCRUAL\_ELIGIBILITY\_VERSION ,*

*from ACCRUAL\_RULE ar, ACCRUAL\_ELIGIBILITY ae*

*where ar.ACCRUAL\_RULE\_ID=ae.ACCRUAL\_RULE\_ID and ar.RECORD\_ACTIVE\_IND='Y' and ae.RECORD\_ACTIVE\_IND='Y' and ae.ELIGIBLE\_IND='Y'*

*and nvl(ae.NO\_OF\_PREV\_MONTHS, 0)>0 and ( ar.PRODUCT\_TYPE\_INTRNL\_ID in (134 , 133 , 131 , 132) ) and ( ar.EFF\_DATE is null or ar.EFF\_DATE<=? ) and nvl(ar.EXP\_DATE, '01-Jan-2999')>=?;*

1. **Get the Eligible TransIds**

*SELECT trans\_id, trans\_version,trans\_type,trans\_stage\_id, trans\_charge\_id,*

*trans\_charge\_version FROM TRANSACTION\_CHARGE\_STAGING TS*

*WHERE accrual\_eligibility\_id =537722 AND accrual\_eligibility\_version=5*

*AND TS.TRADE\_DATE*

*BETWEEN '01-May-2014'*

*AND '31-May-2014'*

*AND record\_status IS NULL*

*ORDER BY trans\_charge\_id,trans\_charge\_version;*

## How to run:

./startNewBatch.sh -DbatchName=JournalAdjustmentDistributor -DbatchType=Monthly -DproductTypes=IRDRV -DstartDate=01/01/2014 -DendDate=01/31/2014

## Post Check

*SELECT COUNT(\*)*

*FROM Trade\_Journal*

*WHERE PARTY\_INTRNL\_ID =43182*

*AND product\_type\_intrnl\_id =131*

*AND TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014';*

*SELECT \**

*FROM transaction\_Journal*

*WHERE PARTY\_INTRNL\_ID =43182*

*AND product\_type\_intrnl\_id =131*

*AND TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014';*

## Debug

## Clean up

*DELETE*

*FROM Trade\_Journal Tj PARTY\_INTRNL\_ID =43182*

*AND product\_type\_intrnl\_id =131*

*AND TC.TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014';*

*DELETE*

*FROM Transaction\_Journal Tj*

*WHERE PARTY\_INTRNL\_ID =43182*

*AND product\_type\_intrnl\_id =131*

*AND TC.TRADE\_DATE BETWEEN '01-Jan-2014' AND '31-Jan-2014';*

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

1. Materiality Check : If the charges are same as that of previous transaction then it will not move records to TransactionJournal or TradeJournal, but it will change the status in Transaction\_charge\_Staging as completed.
2. Depends on the TRANS\_TYPE in Transaction\_Charge\_Staging table, it will move records to either Transaction\_journal or Trade\_journal.

# Manual Accrual to Journal Summary Batch:

## Pre-requisite

Manual Accruals in verified status

## How to run:

Case Run tab: Select Product Type [] Operation Name [ManualAccrualToAllocation] Posting Period [MM-YYYY] and click on Run

## Post Check

*Select \* from Journal\_Summary where party\_intrnl\_id = ? and Posting\_period = ‘MM-YYYY’*

## Debug

Verify Journal Summary Key is set up for the Product Type and Charge Type

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

# Transaction Journal to Journal Summary Batch:

## Pre-requisite

## How to run:

./startNewBatch.sh -DstartNewBatch=true -DbatchName=SummaryDistributor -DbatchType=Daily -DsummaryType=JournalSummary -DjournalType=TransactionJournalToJournlSummary -DproductTypes=FX -DpostingPeriod=12-2014

## Post Check

*Select \* from Journal\_Summary where party\_intrn\_id = ? and Posting\_period = ‘MM-YYYY’*

## Debug

Verify that the Party is set up for Invoice Recon Method = Summary

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

# Trade Journal to Journal Summary Batch:

## Pre-requisite

## How to run:

./startNewBatch.sh -DbatchName=SummaryDistributor -DbatchType=Monthly -DsummaryType=JournalSummary -DjournalType=TradeJournalToJournlSummary -DproductTypes=IRDRV -DpostingPeriod=01-2013

## Post Check

*Select \* from Journal\_Summary where party\_intrn\_id = ? and Posting\_period = ‘MM-YYYY’*

## Debug

Verify that the Party is set up for Invoice Recon Method = Transaction

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

# Expense Summary Batch:

## Pre-requisite

1. Summary key (ELCSummary) should have configured for Product and Charge. Posting period should fall under the effective and expiry date of the Summary key

*SELECT DISTINCT sk.\**

*FROM SUMMARY\_KEY sk*

*WHERE sk.RECORD\_ACTIVE\_IND = 'Y'*

*AND sk.SUMM\_KEY\_TYPE = 'ELCSummary'*

*AND EXISTS*

*(SELECT 1*

*FROM JOURNAL\_SUMMARY js*

*WHERE js.PROCESS\_STATE\_CD IS NULL*

*AND js.RECORD\_ACTIVE\_IND = 'Y'*

*AND sk.PRODUCT\_TYPE\_INTRNL\_ID = js.PRODUCT\_TYPE\_INTRNL\_ID*

*AND sk.CHARGE\_TYPE\_INTRNL\_ID = js.CHARGE\_TYPE\_INTRNL\_ID*

*AND sk.EFF\_DATE <= TO\_DATE('01-' || js.POSTING\_PERIOD,'DD-MM-YYYY')*

*AND ( sk.EXP\_DATE IS NULL*

*OR sk.EXP\_DATE >= TO\_DATE('01-'|| js.POSTING\_PERIOD,'DD-MM-YYYY') )*

*) AND sk.PRODUCT\_TYPE\_INTRNL\_ID IN (121,122 );*

## How to run:

1. **Run Command**

./startNewBatch.sh -DstartNewBatch=true -DbatchName=SummaryDistributor -DbatchType=$batchType -DsummaryType=ELCSummary -DproductTypes=$productType

1. **You can also run this batch from case run:**

**ManualAccrualToAllocation**

## Post Check

**After this command ran, it will retrieve the eligible JOURNAL\_SUMMARY using following query**

*SELECT \**

*FROM JOURNAL\_SUMMARY journalsum0\_*

*WHERE journalsum0\_.RECORD\_ACTIVE\_IND ='Y'*

*AND ( journalsum0\_.PROCESS\_STATE\_CD IS NULL )*

*AND to\_date('01-'*

*||posting\_period, 'dd-mm-yyyy')>=?*

*AND to\_date('01-'*

*||posting\_period, 'dd-mm-yyyy') <=NVL(?, '31-Dec-9999')*

*AND journalsum0\_.PRODUCT\_TYPE\_INTRNL\_ID=?*

*AND journalsum0\_.CHARGE\_TYPE\_INTRNL\_ID =?;*

1. EXPENSE\_LINE\_SUMMARY should have generated.
2. JOURNAL\_SUMMARY should have updated as PROCESS\_STATE\_CD = ‘ELCSummary’
3. POSTING\_AMOUNT of JS and ELC for the product, charge, party, posting period should be same.

**JS is summarised in to ELS based on below query:** Here it prepares the group by based on key setup

*SELECT SUM(GROSS\_FEE) grossFee,*

*SUM(GROSS\_FEE\_IN\_BASE\_CURR) grossFeeInBaseCurr,*

*SUM(INDICATIVE\_FEE) indicativeFee,*

*SUM(INDICATIVE\_FEE\_IN\_BASE\_CURR) indicativeFeeInBaseCurr,*

*SUM(NET\_FEE) netFee ,*

*SUM(NET\_FEE\_IN\_BASE\_CURR) netFeeInBaseCurr,*

*SUM(NOTIONAL) notional ,*

*SUM(NOTIONAL\_IN\_BASE\_CURR) notionalInBaseCurr,*

*SUM(POSTING\_AMOUNT) postingAmount,*

*SUM(POSTING\_AMOUNT\_IN\_BASE\_CURR) postingAmountInBaseCurr,*

*SUM(REBATE) rebate,*

*SUM(REBATE\_IN\_BASE\_CURR) rebateInBaseCurr,*

*SUM(SURCHARGE) surcharge ,*

*SUM(SURCHARGE\_IN\_BASE\_CURR) surchargeInBaseCurr,*

*SUM(DISCOUNT) discount ,*

*SUM(DISCOUNT\_IN\_BASE\_CURR) discountInBaseCurr,*

*SUM(ESTIMATED\_FEE) estimatedFee ,*

*SUM(ESTIMATED\_FEE\_IN\_BASE\_CURR) estimatedFeeInBaseCurr,*

*SUM(TAX) tax,*

*SUM(TAX\_IN\_BASE\_CURR) taxInBaseCurr,*

*SUM(QUANTITY)quantity,*

*SUM(TRANS\_ACTIVITY\_COUNT) transActivityCount,*

*SUM(trans\_count) transCount,*

*BASE\_CURR baseCurr,*

*CHARGE\_TYPE\_INTRNL\_ID chargeTypeIntrnlId,*

*DEAL\_TYPE dealType,*

*EXEC\_METHOD execMethod,*

*EXPENSE\_ACCT\_INTRNL\_ID expenseAcctIntrnlId,*

*INSTR\_TYPE\_INTRNL\_ID instrTypeIntrnlId,*

*INVOICE\_LABEL\_ID invoiceLabelId,*

*LEGAL\_ENTITY\_INTRNL\_ID legalEntityIntrnlId,*

*PARTY\_INTRNL\_ID partyIntrnlId,*

*PAYABLE\_ACCT\_INTRNL\_ID payableAcctIntrnlId,*

*POSTING\_CURR postingCurr,*

*PRODUCT\_TYPE\_INTRNL\_ID productTypeIntrnlId ,*

*JOURNAL\_TYPE journalType,*

*POSTING\_VALUE\_DATE postingValueDate,*

*BUSINESS\_DATE businessDate,*

*POSTING\_PERIOD postingPeriod,*

*EXPENSE\_LINK\_ID expenseLinkId*

*FROM JOURNAL\_SUMMARY*

*WHERE PROCESS\_STATE\_CD IS NULL*

*AND RECORD\_ACTIVE\_IND ='Y'*

*AND PRODUCT\_TYPE\_INTRNL\_ID =?*

*AND CHARGE\_TYPE\_INTRNL\_ID = ?*

*AND to\_date('01-'*

*||posting\_period,'dd-mm-yyyy') >= ?*

*AND to\_date('01-'*

*||posting\_period,'dd-mm-yyyy') <= NVL(?,'31-Dec-9999')*

*GROUP BY BASE\_CURR,*

*CHARGE\_TYPE\_INTRNL\_ID,*

*DEAL\_TYPE,*

*EXEC\_METHOD,*

*EXPENSE\_ACCT\_INTRNL\_ID,*

*INSTR\_TYPE\_INTRNL\_ID,*

*INVOICE\_LABEL\_ID,*

*LEGAL\_ENTITY\_INTRNL\_ID,*

*PARTY\_INTRNL\_ID,*

*PAYABLE\_ACCT\_INTRNL\_ID,*

*POSTING\_CURR,*

*PRODUCT\_TYPE\_INTRNL\_ID ,*

*JOURNAL\_TYPE,*

*POSTING\_VALUE\_DATE,*

*BUSINESS\_DATE,*

*POSTING\_PERIOD,*

*EXPENSE\_LINK\_ID*

## Debug

## Clean up

1. Update the respective JOURNAL\_SUMMARY to process\_state\_cd = null
2. Delete the respective EXPENSE\_LINE\_SUMMARY records.
3. Run the batch command again

### Scenario 1

If Summary key (ELCSummary) is not set up.

#### Solution

Run the query from pre-requisite. It will show whether the summary key is configured or not.

If not go on following path and create summary key:

AAP->Accrual->SummaryKey

### Scenario 2

#### Solution

## Configurations / Points to Note

1. SummaryKey configuration SUMMARY\_KEY

*select \* from SUMMARY\_KEY where SUMM\_KEY\_TYPE = 'ELCSummary' and PRODUCT\_TYPE\_INTRNL\_ID = 101 and CHARGE\_TYPE\_INTRNL\_ID = 201;*

1. Link attribute (Hard key)

*select \* from ACCRUAL\_LINK\_REFERENCE where LINK\_TYPE ='ELCSummaryKey';*

*select \* from ACCRUAL\_LINK\_ATTRIBUTE where LINK\_REF\_NO = 200 ;*

# GL Enrichment Batch:

## Pre-requisite

1. Expense line summaries should have generated. It picks up the eligible ELC using below query:

*SELECT elc.EXPENSE\_LINE\_SUMM\_ID,*

*elc.EXPENSE\_LINE\_SUMM\_VERSION*

*FROM Expense\_Line\_Summary elc*

*WHERE elc.ALLOCATION\_STATUS IS NULL*

*AND elc.EXCEPTION\_IND = 'N'*

*AND elc.RECORD\_ACTIVE\_IND = 'Y' ;*

1. Account mapping rule should have created for product charge and appropriate payable and expense account should have been set to the account mapping.

*SELECT \**

*FROM ACCOUNT\_MAPPING\_RULE accountmap0\_*

*WHERE accountmap0\_.CHARGE\_TYPE\_CODE ='Brokerage'*

*AND accountmap0\_.RECORD\_ACTIVE\_IND ='Y'*

*AND ( accountmap0\_.EFF\_DATE IS NULL*

*OR accountmap0\_.EFF\_DATE <='01-Nov-2014' )*

*AND NVL(accountmap0\_.EXP\_DATE, '01-Jan-3000')>='01-Nov-2014'*

*AND accountmap0\_.PRODUCT\_TYPE\_CODE ='FX'*

*ORDER BY accountmap0\_.PRIORITY DESC;*

## How to run:

1. Command

./startNewBatch.sh -DstartNewBatch=true -DbatchName=GLAccountEnrichment -DbatchType=$batchType -DproductTypes=$productType

1. CASE Run:

ManualAccrualToAllocation

## Post Check

1. In Expense\_Line\_Summary table the EXPENSE\_ACCT\_INTRNL\_ID, PAYABLE\_ACCT\_INTRNL\_ID, ACCT\_MAPPING\_RULE\_ID, ACCT\_MAPPING\_RULE\_VERSION, ALLOCATION\_STATUS should have populated and ALLOCATION\_STATUS should be ‘Pending Allocation’.

*SELECT elc.EXPENSE\_LINE\_SUMM\_ID, elc.EXPENSE\_LINE\_SUMM\_VERSION,elc.EXPENSE\_ACCT\_INTRNL\_ID,elc.PAYABLE\_ACCT\_INTRNL\_ID,ACCT\_MAPPING\_RULE\_ID,ACCT\_MAPPING\_RULE\_VERSION,ALLOCATION\_STATUS*

*FROM Expense\_Line\_Summary elc*

*WHERE elc.ALLOCATION\_STATUS IS NULL*

*AND elc.EXCEPTION\_IND = 'N'*

*AND elc.RECORD\_ACTIVE\_IND = 'Y' ;*

## Debug

## Clean up

1. Allocation status of the EXPENSE\_LINE\_SUMM should be marked as null.

### Scenario 1

1. If account mapping rule not found it will not process the ELC and throws AccountMappingRuleNotFound.
2. If two or more account mapping with same priority found will throw ***MultipleAccountMappingRules.***

#### Solution

Set the correct account mapping rule and appropriate payable and expense account.

### Scenario 2

#### Solution

## Configurations / Points to Note

# Allocation Batch:

## Pre-requisite

1. GL Enrichment batch should have run before running Allocation batch.

Allocation batch picks up ELC using below query

*SELECT expenselin0\_.EXPENSE\_LINE\_SUMM\_ID AS col\_0\_0\_,*

*expenselin0\_.EXPENSE\_LINE\_SUMM\_VERSION AS col\_1\_0\_,*

*expenselin0\_.CHARGE\_TYPE\_INTRNL\_ID AS col\_2\_0\_,*

*expenselin0\_.PARTY\_INTRNL\_ID AS col\_3\_0\_,*

*expenselin0\_.PRODUCT\_TYPE\_INTRNL\_ID AS col\_4\_0\_*

*FROM EXPENSE\_LINE\_SUMMARY expenselin0\_*

*WHERE (expenselin0\_.ALLOCATION\_STATUS ='Pending Allocation' )*

*AND(expenselin0\_.RECORD\_ACTIVE\_IND ='Y' )*

*AND(expenselin0\_.EXCEPTION\_IND ='N' )*

*AND(expenselin0\_.POSTING\_PERIOD ='03-2015' )*

*AND(expenselin0\_.PRODUCT\_TYPE\_INTRNL\_ID IN(101 , 103 , 102));*

In process batch:

1. Batch does STP check, it check ELC has limit rule id and version.- Not using right now.

2. Validation for ELC. POSTING\_AMOUNT\_IN\_BASE\_CURR, ELC is paid, ELC. POSTING\_VALUE\_DATE

3. Checks has any overriding

*select \* from MANUAL\_ALLOCATION\_OVERRIDE manualallo0\_*

*where manualallo0\_.EXPENSE\_LINE\_SUMM\_ID=? and manualallo0\_.POSTING\_PERIOD=?*

4.If above validation passed it gets the allocation rule:

*select \* from ALLOCATION\_RULE model*

*where model.CHARGE\_TYPE\_CODE = 'Brokerage'*

*and model.PRODUCT\_TYPE\_CODE = 'FX'*

*and model.EFF\_DATE <= '01-Mar-2015'*

*and ( model.EXP\_DATE is null or model.EXP\_DATE >= '01-Mar-2027' )*

*and model.RECORD\_ACTIVE\_IND='Y'*

*order by model.priority desc;*

5.Based on journalSummary.ruleBasisValues and alloction.ruleBasis, it finds out applicable allocation rule.

6.It does the allocation based on allocation sub type i.e. fixed, activity, Percentage based allocation.

7. Puts entry in Summary Allocation.

## How to run:

1. Command

./startNewBatch.sh -DstartNewBatch=true -DbatchName= AllocationDistributor -DbatchType=$batchType -DproductTypes=$productType –DpostingPeriod=$postingPeriod

1. CASE Run:

ManualAccrualToAllocation

## Post Check

1. ELS should have allocation\_status = Allocated
2. SUMMARY\_ALLOCATION should have entries.

## Debug

## Clean up

### Scenario 1

#### Solution

### Scenario 2

#### Solution

## Configurations / Points to Note

# Postings Batch:

## Pre-requisite

*select \* from Summary\_Allocation sa where sa.POSTING\_PERIOD ='01-2014' and party\_Intrnl\_Id=7259920 and sa.POSTING\_STATUS is null;*

## How to run:

./startNewBatch.sh -DbatchName=GLPostingDistributor -DbatchType=Daily -DproductTypes=BANCKCHARGE -DpostingPeroid=01-2014

## Post Check

*select \* from Summary\_Allocation sa where sa.POSTING\_PERIOD ='01-2014' and party\_Intrnl\_Id=7259920 and sa.POSTING\_STATUS is not null;*

*select \* from GL\_POSTING where POSTING\_PERIOD='01-2014' and PARTY\_INTRNL\_ID=(7259920);*

## Debug

1. Check whether POSTING\_STATUS is not null if batch ran successfully for Posting Peroid and PartyINTRNLID.

## Clean up

*update Summary\_Allocation sa set POSTING\_STATUS=null where sa.POSTING\_PERIOD ='01-2014' and party\_Intrnl\_Id=7259920 and sa.POSTING\_STATUS is not null;*

GLPOSTING

*delete from GL\_POSTING where POSTING\_PERIOD='01-2014' and PARTY\_INTRNL\_ID=(7259920);*

### Scenario 1 :

### Creating ELC Approved Records and Running Posting Batch for Period and Product without cleaning of Already Posted Records

#### Solution

### Scenario 2

### Creating ELC Approved Records and Running Posting Batch for Peroid and Product cleaning of Already Posted Records

#### Solution

## Configurations / Points to Note

# Trade Line Detail Batch:

## Pre-requisite

*select \* from trade\_line\_detail where POSTING\_PERIOD = '04-2015' and PARTY\_INTRNL\_ID =722133 and product\_type\_intrnl\_id =131;*

## How to run:

./startNewBatch.sh -DbatchName=TradeAccrualDistributor -DproductTypes=IRDRV -DpostingPeriod=03-2015 -DbatchType=Daily

## Post Check

*SELECT NVL(SUM(TJ.DISCOUNT),0) discount,*

*NVL(SUM(TJ.ESTIMATED\_FEE),0) estimatedFee,*

*NVL(SUM(TJ.ESTIMATED\_FEE\_IN\_BASE\_CURR),0) estimatedFeeInBaseCurr,*

*NVL(SUM(TJ.GROSS\_FEE),0) grossFee,*

*NVL(SUM(TJ.GROSS\_FEE\_IN\_BASE\_CURR),0) grossFeeInBaseCurr,*

*NVL(SUM(TJ.INDICATIVE\_FEE),0) indicativeFee,*

*NVL(SUM(TJ.INDICATIVE\_FEE\_IN\_BASE\_CURR),0) indicativeFeeInBaseCurr,*

*NVL(SUM(TJ.NET\_FEE),0) netFee,*

*NVL(SUM(TJ.NET\_FEE\_IN\_BASE\_CURR),0) netFeeInBaseCurr,*

*NVL(SUM(TJ.NOTIONAL),0) notional,*

*NVL(SUM(TJ.PAID\_AMOUNT),0) paidAmount,*

*NVL(SUM(TJ.POSTING\_AMOUNT),0) postingAmount,*

*NVL(SUM(TJ.POSTING\_AMOUNT\*TJ.POSTING\_CURR\_FX\_RATE),0) postingAmountInBaseCurr,*

*NVL(SUM(TJ.QUANTITY),0) quantity,*

*NVL(SUM(TJ.REBATE),0) rebate,*

*NVL(SUM(TJ.REBATE\_IN\_BASE\_CURR),0) rebateInBaseCurr,*

*NVL(SUM(TJ.SURCHARGE),0) surcharge,*

*NVL(SUM(TJ.SURCHARGE\_IN\_BASE\_CURR),0) surchargeInBaseCurr,*

*NVL(SUM(TJ.SURCHARGE\_PAID),0) surchargePaid,*

*NVL(SUM(TJ.TAX),0) tax,*

*NVL(SUM(TJ.TAX\_IN\_BASE\_CURR),0) taxInBaseCurr,*

*NVL(SUM(TJ.TAX\_PAID),0) taxPaid,*

*TJ.CHARGE\_TYPE\_INTRNL\_ID chargeTypeIntrnlId,*

*TJ.TRANS\_ID transId*

*-- TJ.JOURNAL\_SUMM\_ID,TJ.JOURNAL\_SUMM\_version,count(\*)*

*FROM TRADE\_JOURNAL TJ,*

*JOURNAL\_SUMMARY JS*

*WHERE TRADE\_LINE\_DETAIL\_ID    IS NULL*

*AND TJ.JOURNAL\_SUMM\_ID         = JS.JOURNAL\_SUMM\_ID*

*AND TJ.JOURNAL\_SUMM\_VERSION    = JS.JOURNAL\_SUMM\_VERSION*

*AND TJ.RECORD\_ACTIVE\_IND       = 'Y'*

*AND JS.RECORD\_ACTIVE\_IND       = 'Y'*

*AND TJ.POSTING\_PERIOD          = '03-2016'*

*AND JS.POSTING\_STATUS          = 'Posted'*

*AND TJ.PARTY\_INTRNL\_ID         =21776306*

*AND TJ.PRODUCT\_TYPE\_INTRNL\_ID IN (121)*

*AND TJ.CHARGE\_TYPE\_INTRNL\_ID  IN (201)*

*GROUP BY TJ.TRANS\_ID,*

*--GROUP BY TJ.JOURNAL\_SUMM\_ID,TJ.JOURNAL\_SUMM\_version,*

*TJ.CHARGE\_TYPE\_INTRNL\_ID;*

## Debug

1)Check Whether Journal\_Summary for Party and Peroid has Posting Peroid has status ‘Posted’.

2)Check whether TRADE\_LINE\_DETAIL\_**ID is populated in TRADE\_JOURNAL** if batch runs succcesfully

3)Check whether JOURNAL\_SUMM\_ID **is populated in TRADE\_JOURNAL** if batch doesn’t populate data in Trade\_Line\_Detail.

4)IF there Is stacktrace like below ,please whether ACCRUAL\_ELIGIBILITY\_ID is Populated in **TRADE\_JOURNAL**.

**java.rmi.RemoteException: org.hibernate.PropertyValueException: not-null property references a null or transient value: com.cs.examin.dataobject.aap.invoicepayment.TradeLineDetail.accrualEligibilityId**

## Clean up

*UPDATE TRADE\_JOURNAL js*

*SET TRADE\_LINE\_DETAIL\_ID=nul l*

*WHERE js.POSTING\_PERIOD = '03-2015'*

*AND js.PARTY\_INTRNL\_ID ='722133';*

*UPDATE trade\_line\_detail*

*SET Record\_ACTIVE\_IND='N'*

*WHERE POSTING\_PERIOD = '03-2015'*

*AND PARTY\_INTRNL\_ID ='722133';*

### Scenario 1

Batch can be Run from Case Run Screen

#### Solution

### Scenario 2

Whenever Invoice is Approved ,internally TLD Batch will be called.

#### Solution

### Scenario 3

Batch can be from Command Line.

#### Solution

## Configurations / Points to Note

1)**Currently When we run batch with parameters productTypes:IRDRV ,partyIds:722133 but still batch runs for all Parties and Products.We need to fix issue.**

# Invoice Accrual Line Detail Batch:

## Pre-requisite

*SELECT \**

*FROM INVOICE\_ACCRUAL\_LINE\_DETAIL*

*WHERE POSTING\_PERIOD='06-2013'*

*AND PARTY\_INTRNL\_ID =7259920;*

## How to run:

./startNewBatch.sh -DbatchName=InvoiceAccrualDistributor -DproductTypes=BANKCHARGE -DpostingPeriod=03-2016 -DbatchType=Daily

## Post Check

*SELECT \**

*FROM INVOICE\_ACCRUAL\_LINE\_DETAIL*

*WHERE POSTING\_PERIOD='06-2013'*

*AND PARTY\_INTRNL\_ID =7259920;*

## Debug

1)Check whether JOURNAL SUMMARY PRESENT FOR PEROID and Party.

2)If IALD is not generated ,check whether Invoice\_Accrual\_Key is present.

3)Still have issues, please check for exceptions in logs or check exception screen

## Clean up

To Re Run IALD Batch update journal\_summary with below command .

Make sure to delete or set RecordActiveIND to N in invoice\_accrual\_line\_detail Table as shown below

Run The below Queries

*UPDATE journal\_summary js*

*SET js.INVOICE\_ACCRUAL\_LINE\_ID = NULL ,*

*js.INVOICE\_ACCRUAL\_LINK\_ID = NULL ,*

*js.INVOICE\_IALD\_ID = NULL ,*

*js.INVOICE\_LINK\_ID = NULL*

*WHERE js.POSTING\_PERIOD = PostingPeroid*

*AND js.party\_intrnl\_id = partyInrID;*

*DELETE*

*FROM INVOICE\_ACCRUAL\_LINE\_DETAIL*

*WHERE POSTING\_PERIOD=:PostingPeroid*

*AND PARTY\_INTRNL\_ID= :partyInrID;*

Or

*UPDATE invoice\_accrual\_line\_detail*

*SET Record\_ACTIVE\_IND=’N’*

*WHERE party\_intrnl\_id=7259920*

*AND posting\_period ='06-2013';*

### Scenario 1 :

Batch can be Run from Case Run Screen

#### Solution

### Scenario 2

Whenever Invoice is Approved, internally IALD Batch will be called

#### Solution

### Scenario 2

Batch Can be Run from Command line.

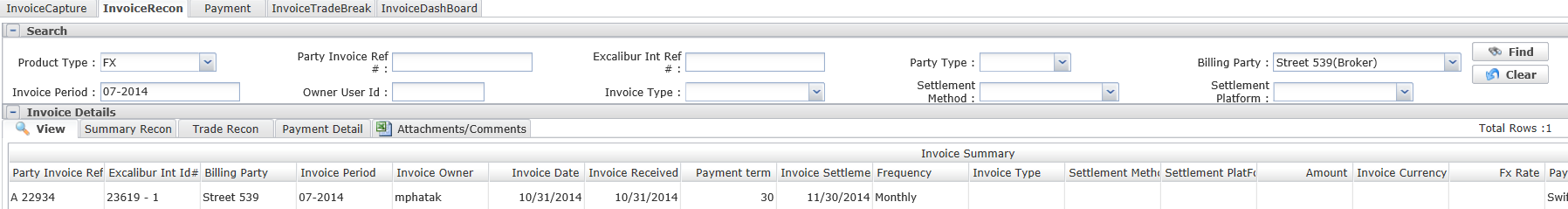
#### Solution

## Configurations / Points to Note

# Invoice Trade Auto-match Batch:

## Pre-requisite

1. Invoice should exist for the Party, Product and Period. Go to the ExcaliburWeb 🡪 Invoice 🡪 InvoiceRecon Tab. Key in the Product, Party, Period and find the invoice. The captured invoices would be displayed. If it does not exist, create the invoice or reach out to Ops team to create an invoice.



To query in the back end

*SELECT ih.\**

*FROM invoice\_header ih,*

*invoice\_detail id*

*WHERE ih.invoice\_id =id.invoice\_id*

*AND ih.record\_active\_ind ='Y'*

*AND id.record\_active\_ind ='Y'*

*AND id.product\_type\_intrnl\_id=?*

*AND (id.party\_intrnl\_id =?*

*OR ih.party\_intrnl\_id =?)*

*AND ih.invoice\_period ='?'*

*And id.recon\_method=’Summary’;*

1. Trade Line Details should exist for the Product, Party and Period. If it does not exist, check for the TLD Batch and debug Options

*SELECT COUNT(\*)*

*FROM Trade\_Line\_Detail tld*

*WHERE tld.party\_intrnl\_id =?*

*AND tld.posting\_period =?*

*AND tld.product\_type\_intrnl\_id=?*

*AND record\_active\_ind ='Y';*

1. Recon Rules should be defined for Product.

*SELECT \**

*FROM recon\_rule rr*

*WHERE rr.rule\_type ='TradeRecon'*

*AND rr.product\_type\_code=?*

*AND rr.record\_active\_ind='Y';*

*parameter example – ‘FX’*

## How to run:

### Method 1

To run from Unix Box or To run from Batch jobs Screen

-DbatchName=MatchingDistributor -DbatchType=Daily -DproductTypes=IRDRV -DInvoicePeriod=12-2015 -DruleType=TradeRecon -DstartNewBatch=true -DuseCache=yes -DInvoiceId=3083

### Method 2

To run from Web

Goto Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go To TradeRecon tab. Click on AutoMatch Button.

## Post Check

### Method 1

To check in database.

*SELECT \**

*FROM invoice\_accrual\_reconciliation*

*WHERE recon\_match\_id IN*

*(SELECT recon\_match\_id*

*FROM invoice\_accrual\_reconciliation iar*

*WHERE iar.party\_intrnl\_id=722133*

*AND iar.recon\_module ='InvoiceTradeRecon'*

*AND iar.record\_active\_ind='Y'*

*AND iar.posting\_period ='05-2014'*

*AND iar.recon\_type ='Auto'*

*);*

### Method 2

To check from Web

Go to Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go to TradeRecon - Matched tab. Select Reconciled, Un-Reconciled Check Boxes and click on Refresh Button.

## Debug

## Clean up

### Scenario 1

#### To Cleanup form Back End

#### Solution

Delete from Invoice Accrual Reconciliation

*Delete*

*FROM invoice\_accrual\_reconciliation*

*WHERE recon\_match\_id IN*

*(SELECT recon\_match\_id*

*FROM invoice\_accrual\_reconciliation iar*

*WHERE iar.party\_intrnl\_id=722133*

*AND iar.recon\_module ='InvoiceTradeRecon'*

*AND iar.record\_active\_ind='Y'*

*AND iar.posting\_period ='05-2014'*

*AND iar.recon\_type ='Auto'*

*);*

Update Trade Line Detail

I have to include adjustment amount reset in this query

*UPDATE trade\_line\_detail tld*

*SET tld.recon\_match\_id =NULL,*

*tld.recon\_status =NULL,*

*update\_user\_id='statusReset',*

*update\_datetime=sysdate,*

*Invoice\_adj\_amount = OUTSTANDING\_ACCRUAL\_AMOUNT*

*WHERE tld.party\_intrnl\_id =12121*

*AND tld.posting\_period ='07-2014'*

*AND tld.product\_type\_intrnl\_id=101*

*AND tld.record\_active\_ind ='Y';*

Update Invoice Transaction

*UPDATE invoice\_transaction*

*SET recon\_status =NULL,*

*update\_user\_id ='statusReset',*

*update\_datetime=sysdate*

*WHERE invoice\_id =12121;*

### Scenario 2

To Cleanup from Web

### Solution

Go to Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go to TradeRecon - Matched tab.

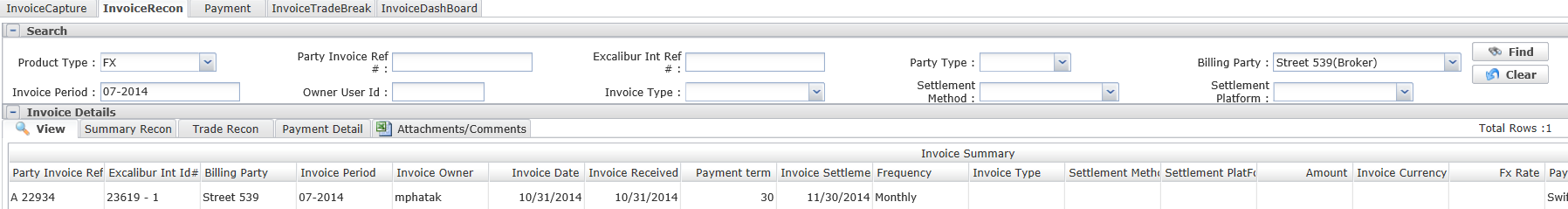
Click on “Select All” check Box. Click on UnReconclie. This would unreconciled records if they are in UnReconciled or Reconciled State

## Configurations / Points to Note

# Invoice Summary Auto-Match Batch:

## Pre-requisite

1. Invoice should exist for the Party, Product and Period. Go to the ExcaliburWeb 🡪 Invoice 🡪 InvoiceRecon Tab. Key in the Product, Party, Period and find the invoice. The captured invoices would be displayed. If it does not exist, create the invoice or reach out to Ops team to create an invoice.



To query in the back end

*SELECT ih.\**

*FROM invoice\_header ih,*

*invoice\_detail id*

*WHERE ih.invoice\_id =id.invoice\_id*

*AND ih.record\_active\_ind ='Y'*

*AND id.record\_active\_ind ='Y'*

*AND id.product\_type\_intrnl\_id=?*

*AND (id.party\_intrnl\_id =?*

*OR ih.party\_intrnl\_id =?)*

*AND ih.invoice\_period ='?'*

*And id.recon\_method=’Summary’;*

1. Invoice Accrual Line Details should exist for the Party, Product, Period. If it does not exist, run the IALD Batch and look for debug options in IALD Batch.

*SELECT COUNT(\*)*

*FROM INVOICE\_ACCRUAL\_LINE\_DETAIL iald*

*WHERE iald.party\_intrnl\_id =?*

*AND iald.posting\_period =?*

*AND iald.product\_type\_intrnl\_id=?*

*AND record\_active\_ind ='Y';*

## How to run:

To run from Web

Goto Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go To SummaryRecon tab. Click on AutoMatch Button.

## Post Check

### Method 1

To check in database.

*SELECT \**

*FROM invoice\_accrual\_reconciliation*

*WHERE recon\_match\_id IN*

*(SELECT recon\_match\_id*

*FROM invoice\_accrual\_reconciliation iar*

*WHERE iar.party\_intrnl\_id=722133*

*AND iar.recon\_module ='InvoiceSummaryRecon'*

*AND iar.record\_active\_ind='Y'*

*AND iar.posting\_period ='05-2014'*

*AND iar.recon\_type ='Auto'*

*);*

### Method 2

To check from Web

Go to Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go to TradeRecon - Matched tab. Select Reconciled, Un-Reconciled Check Boxes and click on Refresh Button.

## Debug

## Clean up

### Scenario 1

#### To Cleanup form Back End

#### Solution

Delete from Invoice Accrual Reconciliation

*Delete*

*FROM invoice\_accrual\_reconciliation*

*WHERE recon\_match\_id IN*

*(SELECT recon\_match\_id*

*FROM invoice\_accrual\_reconciliation iar*

*WHERE iar.party\_intrnl\_id=722133*

*AND iar.recon\_module ='InvoiceSummaryRecon'*

*AND iar.record\_active\_ind='Y'*

*AND iar.posting\_period ='05-2014'*

*AND iar.recon\_type ='Auto'*

*);*

Update Invoice Accrual Line Detail

I have to include adjustment amount reset in this query

*UPDATE invoice\_accrual\_line\_detail iald*

*SET recon\_status=null,*

*Invoice\_adj\_amount = OUTSTANDING\_ACCRUAL\_AMOUNT*

*Where party\_intrnl\_id=722133*

*And iar.posting\_period ='05-2014'*

*And recon\_status is not null;*

Update Invoice Invoice Line Detail

*UPDATE invoice\_line\_detail*

*SET payment\_status =NULL,*

*update\_user\_id ='statusReset',*

*update\_datetime=sysdate*

*WHERE invoice\_id =12121;*

### Scenario 2

To Cleanup from Web

### Solution

Go to Invoice 🡪 InvoiceRecon Tab. Search for the invoice and Go to SummaryRecon - Matched tab.

Click on UnReconclie. This would unreconcile records if they are in UnReconciled or Reconciled State

## Configurations / Points to Note

# Charge Clearance Queries By Party

--Step 1

delete from limit\_Tier\_running\_total where running\_Total\_id in (select running\_total\_id from running\_total\_rule where party\_intrnl\_id in (3819)) ;

--Step 2

delete from charge\_running\_total where running\_Total\_id in (select running\_total\_id from running\_total\_rule where party\_intrnl\_id in (3819)) ;

--Step 3

delete /\*+ parallel(20) \*/ from transaction\_charge\_rate where (Trans\_charge\_id,trans\_charge\_Version) in (select /\*+ parallel(20) \*/ trans\_charge\_id,trans\_charge\_version from transaction\_charge where party\_intrnl\_id in (3819) and product\_type\_intrnl\_id in (101,102,103));

--Step 4

delete /\*+ parallel(20) \*/ from transaction\_limit\_rate where (Trans\_charge\_id,trans\_charge\_Version) in (select /\*+ parallel(20) \*/ trans\_charge\_id,trans\_charge\_version from transaction\_charge where party\_intrnl\_id in (3819) and product\_type\_intrnl\_id in (101,102,103));

--Step 5

delete /\*+ parallel(20) \*/ from transaction\_charge where party\_intrnl\_id in (3819) and product\_type\_intrnl\_id in (101,102,103) --and trade\_Date between '01-Dec-2015' and '31-Dec-2015';

--Step 6

delete /\*+ parallel(30) \*/ exception\_details ed where ed.exception\_id in (select /\*+ parallel(30) \*/ exception\_id from examin\_Exception ee where (ee.trans\_id , ee.trans\_Version) in (select /\*+ parallel(30) \*/ et.trans\_id,et.trans\_Version from examin\_transaction et where product\_type\_intrnl\_id in (101,102,103) and trans\_active\_ind ='Y' and calc\_eligible\_ind='Y' and broker\_intrnl\_id in (3819)));

--Step 7

delete from /\*+ parallel(30) \*/ examin\_Exception ee where (ee.trans\_id , ee.trans\_Version) in (select /\*+ parallel(30) \*/ et.trans\_id,et.trans\_Version from examin\_transaction et where product\_type\_intrnl\_id in (101,102,103) and trans\_active\_ind ='Y' and calc\_eligible\_ind='Y' and broker\_intrnl\_id in (3819));

--Step 8

update /\*+ parallel(30) \*/ transaction\_reference tr set trans\_Exception\_ind='N' where (tr.trans\_id, tr.trans\_Version) in (select /\*+ parallel(30) \*/ et.trans\_id,et.trans\_Version from examin\_transaction et where product\_type\_intrnl\_id in (101,102,103) and trans\_active\_ind ='Y' and calc\_eligible\_ind='Y' and broker\_intrnl\_id in (3819));

--Step 9

update /\*+ parallel(30) \*/ examin\_transaction et set calc\_ind='N' where product\_type\_intrnl\_id in (101,102,103) and trans\_active\_ind ='Y' and calc\_eligible\_ind='Y' and broker\_intrnl\_id in (3819);

--Step 10

Update EXAMIN\_BUSINESS\_DATE set date\_value=’01-Dec-2015’ , num\_business\_day =1 where date\_type=’CurrentDate’;

Update EXAMIN\_BUSINESS\_DATE set date\_value=’02-Dec-2015’ where date\_type=’NextDate’;

Update EXAMIN\_BUSINESS\_DATE set date\_value=’30-Nov-2015’ where date\_type=’PreviousDate’;

Update EXAMIN\_BUSINESS\_DATE set date\_value=’31-Dec-2015’ where date\_type=’CurrentMonth’;

Update EXAMIN\_BUSINESS\_DATE set date\_value=’30-Nov-2015’ where date\_type=’PreviousMonth’;

Update EXAMIN\_BUSINESS\_DATE set date\_value=’31-Jan-2016’ where date\_type=’NextMonth’;

Below 2 line should be executed between loop of 20 or 23 days based on number of business days of the month.

./dailyNonEQCalc.sh

./businessDateUpdate.sh

# Accrual Clearance Queries By Party

## Find the count of data and note down

*select count(\*) from invoice\_accrual\_line\_Detail where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from gl\_posting where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from summary\_allocation where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from expense\_line\_summary els where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from journal\_summary where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from transaction\_journal where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*select count(\*) from transaction\_charge\_staging where to\_char(trade\_date, 'MM-YYYY')='12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

**Cleanup Queries**

*delete from invoice\_accrual\_line\_Detail where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from gl\_posting where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from summary\_allocation where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from journal\_summary where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from expense\_line\_summary els where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from transaction\_journal where POSTING\_PERIOD = '12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

*delete from transaction\_charge\_staging where to\_char(trade\_date, 'MM-YYYY')='12-2014' and PARTY\_INTRNL\_ID =43200 and product\_type\_intrnl\_id =101;*

## For Trade Based Accruals

Following two tables are different for trade based accruals

invoice\_accrual\_line\_Detail is replaced by Trade\_Line\_Detail

Transaction\_Jounral is replaced by Trade\_Journal

### Checking Count

*select count(\*) from Trade\_Line\_Detail tld where tld.party\_intrnl\_id=12121 and tld.posting\_period='07-2014' and tld.product\_type\_intrnl\_id=101 and record\_active\_ind='Y';*

*select count(\*) from Trade\_journal tj where tj.party\_intrnl\_id=12121 and tj.posting\_period='07-2014' and tj.product\_type\_intrnl\_id=101 and record\_active\_ind='Y';*

### Cleanup Queries

*delete from Trade\_Line\_Detail tld where tld.party\_intrnl\_id=12121 and tld.posting\_period='07-2014' and tld.product\_type\_intrnl\_id=101 and record\_active\_ind='Y';*

*delete from Trade\_journal tj where tj.party\_intrnl\_id=12121 and tj.posting\_period='07-2014' and tj.product\_type\_intrnl\_id=101 and record\_active\_ind='Y';*

# Invoice Clearance Queries

To clean up a particular invoice

Have a count of records

*select \* from Invoice\_header where invoice\_id=12121;*

*select \* from Invoice\_billing\_summary where invoice\_id=12121;*

*select \* from Invoice\_detail where invoice\_id=12121;*

*select \* from Invoice\_line\_detail where invoice\_id=12121;*

For Transaction Based Invoices

*select \* from invoice\_transaction where invoice\_id=12121;*

*select \* form FX\_invoice\_transaction where invoice\_id=12121;*

Satellite Tables may differ based on product Types

*FX\_invoice\_transaction*

*EQDRV\_invoice\_transaction*

*CRDRV\_invoice\_transaction*

*IRDRV\_invoice\_transaction*

*FUTURES\_invoice\_transaction*

## Clean Up Queries

*delete from Invoice\_header where invoice\_id=12121;*

*delete from Invoice\_billing\_summary where invoice\_id=12121;*

*delete from Invoice\_detail where invoice\_id=12121;*

*delete from Invoice\_line\_detail where invoice\_id=12121;*

For Transaction Based invoices

*delete from invoice\_transaction where invoice\_id=12121;*

*delete form FX\_invoice\_transaction where invoice\_id=12121;*

# Trade Charges

## Search Query for CRDRV



## Search Query for FX



# Recurring Accrual Batch

# LEO Matching

# Look Back / Cap Discount

# Strategy Finder

# Pre-Invoice Distributor / Auto Invoice Capture

# Futures Batch