|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改日期 | 版本 | 修改原因 | 修改人姓名 | 立案單號 |
| 2020/01/03 | 1 | Created  電腦作業申請書191225001833\_理賠情報巡檢表 | 楊智偉 | 200107000521 |
| 2020/05/06 | 2 | 問題單號:20191114-0014  2020/03/03項目A\_1~A\_5改成針對SYS\_NO =A合計項目檢查  2020/05/06 20200224(20200506重新採用). 結案天數改成資料週期是日的要次月2號之後才進行檢查, 月跟年的資料每天進行檢查 | 楊智偉 | 191114000671 |

**一、程式功能概述**

|  |  |  |
| --- | --- | --- |
| 程式功能 | 高管儀表板\_理賠情報巡檢 | |
| 程式名稱 | AAH5\_B332.java | |
| 作業方式 | BATCH | |
| 概要說明 | 高管儀表板\_理賠情報資料巡檢 | |
| 需求單位 | 資料整合資產組 | |
| 作業單位 | 壽險資訊二科 | |
| 作業平台 | ■一般  □平板電腦  □手機 | |
| 使用對象 | ■員工(UCBean)  □客戶(CustomerBean) | |
| **寄信處理** | 寄信對象 | ■無 □客戶　□壽險員工　□關係企業員工　□合作廠商 |
| 寄信方式 | ■無 □Billhunter　□MailSender |

**二、程式流程圖**

巡檢作業檔案

DTZZM010

DBGD.DTGDC009

高管儀表板\_理賠情報巡檢

**三、相關檔案**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 項次 | 中文說明 | 檔案名稱 | **查詢** | **新增** | **修改** | **刪除** |
| 1. | 高管儀表板資料檔 | DTGDC009 | ■ | □ | □ | □ |

**四、相關模組**

|  |  |  |
| --- | --- | --- |
| 項次 | 中文說明 | 程式名稱 |
|  | 異常訊息記錄模組 | batch.ErrorLog |
|  | 批次作業件數記錄模組 | batch.CountManager |
|  | ShutdownDay模組 | com.cathay.common.util.ShutdownDate |
|  | 件數登入模組 | ZZ\_M0Z020 |
|  | 驗證模組 | ZZ\_M4Z001 |

**五、批次基本資料**

|  |  |
| --- | --- |
| 作業名稱 | 高管儀表板\_理賠情報 |
| 業務別 | AA |
| 次系統名稱 | H5 |
| 處理週期 | 日 |
| 分批處理的件數 | 100 |

**六、傳入參數**

|  |  |  |  |
| --- | --- | --- | --- |
| 項次 | 說明 | 資料型態 | 備註 |
|  | STR\_DATE 開始日期 | String | YYYY-MM-DD 可傳可不傳 |
|  | END\_DATE結束日期 | String | YYYY-MM-DD可傳可不傳 |
|  | STAT\_TYPE統計週期 | String | 可傳可不傳  日0  月1  年2 |
|  | SYS\_NO業務別 | String | 可傳可不傳  學團險: H  個險: 0  團險: G  全部: A |
|  | CLAM\_DIV\_NO核賠單位 | String | 可傳可不傳 |

**七、各項交易計算快速連結**

**八、程式內容**

1. 初始：
   1. Set $WORK\_DATE = 系統日期
   2. 取得系統SHUTDOWN DAY： CALL CathayDate模組取得系統SHUTDOWN DAY：
   3. 件數歸零：
      1. START、
      2. 輸入件數、
      3. A\_1錯誤件數、
      4. A\_2錯誤件數、
      5. A\_3錯誤件數、
      6. A\_4錯誤件數、
      7. A\_5錯誤件數、
      8. B\_1錯誤件數、
      9. C\_1錯誤件數、
      10. C\_2錯誤件數、
      11. C\_3錯誤件數、
      12. C\_4錯誤件數、
      13. C\_5錯誤件數、
      14. C\_6錯誤件數
   4. 查詢條件變數 : 都先給null 初始化
      1. $STR\_DATE 開始日期
      2. $END\_DATE結束日期
      3. $STAT\_TYPE統計週期
      4. $SYS\_NO業務別
      5. $CLAM\_DIV\_NO核賠單位
   5. 存放各類巡檢明細LIST物件
      1. A\_1\_LIST, A\_2\_LIST, A\_3\_LIST, A\_4\_LIST, A\_5\_LIST,
      2. B\_1\_LIST,
      3. C\_1\_LIST, C\_2\_LIST, C\_3\_LIST, C\_4\_LIST, C\_5\_LIST, C\_6\_LIST
   6. 將各類巡檢明細LIST物件 塞入第一筆欄位名稱
      1. A\_1\_LIST, A\_2\_LIST, A\_3\_LIST, A\_4\_LIST, A\_5\_LIST 都是 塞下列表頭
      2. SET $HEAD\_LIST = 空的LIST
      3. $HEAD\_LIST 塞入 ‘資料日期’
      4. $HEAD\_LIST 塞入 ‘週期’
      5. $HEAD\_LIST 塞入 ‘業務別’
      6. $HEAD\_LIST 塞入 ‘核賠單位’
      7. $HEAD\_LIST 塞入 ‘數字’
      8. A\_1\_LIST, A\_2\_LIST, A\_3\_LIST, A\_4\_LIST, A\_5\_LIST 塞入 $HEAD\_LIST
   7. B\_1\_LIST 塞下列表頭
      1. SET $HEAD\_LIST = 空的LIST
      2. $HEAD\_LIST 塞入 ‘資料日期’
      3. $HEAD\_LIST 塞入 ‘週期’
      4. $HEAD\_LIST 塞入 ‘業務別’
      5. $HEAD\_LIST 塞入 ‘核賠單位’
      6. $HEAD\_LIST 塞入 ‘業務別’
      7. B\_1\_LIST 塞入 $HEAD\_LIST
   8. C\_1\_LIST, C\_2\_LIST, C\_3\_LIST, C\_4\_LIST, C\_5\_LIST都是 塞下列表頭
      1. SET $HEAD\_LIST = 空的LIST
      2. $HEAD\_LIST 塞入 ‘資料日期’
      3. $HEAD\_LIST 塞入 ‘週期’
      4. $HEAD\_LIST 塞入 核賠單位’
      5. $HEAD\_LIST 塞入 ‘全部’
      6. $HEAD\_LIST 塞入 ‘學團險’
      7. $HEAD\_LIST 塞入 ‘個險’
      8. $HEAD\_LIST 塞入 ‘團險’
      9. C\_1\_LIST, C\_2\_LIST, C\_3\_LIST, C\_4\_LIST, C\_5\_LIST 塞入 $HEAD\_LIST
   9. C\_6\_LIST都是 塞下列表頭
      1. $HEAD\_LIST 塞入 ‘資料日期’
      2. $HEAD\_LIST 塞入 ‘週期’
      3. $HEAD\_LIST 塞入 核賠單位’
      4. $HEAD\_LIST 塞入 ‘數字’
      5. C\_6\_LIST 塞入 $HEAD\_LIST
   10. 先寫入一筆CountManager，COUNT\_NAME＝”START”，COUNT＝0。【可參考程式AKB5\_B800.java↓】

private void initCountManager() throws ModuleException {

countManager.createCountType("START");

countManager.writeLog();

countManager.clearCountTypeAndNumber();

countManager.createCountType(INPUT\_COUNT);

countManager.createCountType(OUTPUT\_COUNT);

countManager.createCountType(ERROR\_COUNT);

countManager.createCountType(DATA\_NOT\_FOUND\_COUNT);

}

1. 傳入參數檢核：
   1. 可無傳入參數：
      1. $STR\_DATE = ＄SHUTDOWN\_DAY，
      2. $END\_DATE = ＄SHUTDOWN\_DAY，
      3. Log.fatal（”無傳入參數”）；
   2. 有傳入參數：
      1. Log.fatal（”有傳入參數, 內容:” + 傳入參數內容） ；
      2. 如果傳入參數個數 < 2
         1. 訊息中文＝” 傳入參數異常”，
         2. 摘　　要＝傳入傳入參數至少要傳STR\_DATE 及 END\_DATE 2個以上 ，
         3. CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，
         4. 錯誤件數++，
         5. RollBack回程式初始狀態資料，結束程式且程式執行結果異常。
      3. 如果傳入參數個數 >=2
         1. $STR\_DATE = 傳入. 開始日期，
         2. $END\_DATE = 傳入. 結束日期，
      4. 如果傳入參數個數 >=3
         1. $ STAT\_TYPE = 傳入. 統計週期
      5. 如果傳入參數個數 >=4
         1. $ SYS\_NO = 傳入.業務別
      6. 如果傳入參數個數 >=5
         1. $ CLAM\_DIV\_NO = 傳入. 核賠單位
   3. 檢核傳入參數，全部檢核完有錯再一次輸出錯誤訊息
      1. SET $錯誤訊息 = 用來存放錯誤訊息的變數
      2. $STR\_DATE 及 $END\_DATE：
         1. 須為西元年格式, 合理日期 且 $STR\_DATE <= $END\_DATE
         2. 不符時, $錯誤訊息 加上 : 開始日期 及 結束日期 須為西元年格式, 合理日期; 且開始日期 <= 結束日期.
      3. $STAT\_TYPE :
         1. 不為空值時, 只能是 0,A,G或H
         2. 不符時, $錯誤訊息 加上 : 統計週期不為空值時, 只能是 0日,1月或2年
      4. $ SYS\_NO
         1. 不為空值時, 只能是 0,1或2
         2. 不符時, $錯誤訊息 加上 : 業務別不為空值時, 只能是 0個險,A全部, G團險或H學團險
      5. $ CLAM\_DIV\_NO
         1. 不為空值時, 須為7碼英數字
         2. 不符時, $錯誤訊息 加上 : 核賠單位不為空值時, 須為7碼英數字
      6. 如果$錯誤訊息 不是空的
         1. Log.fatal（“傳入參數異常: “＋$錯誤訊息），
         2. 訊息中文＝“傳入參數異常”，
         3. 摘　　要＝$錯誤訊息，
         4. CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，
         5. 錯誤件數++，
         6. RollBack回程式初始狀態資料，結束程式且程式執行結果異常。
2. 業務說明 : 將指定日期區間之內 (其他選擇性條件為統計週期, 業務別 及核賠單位), 符合條件的DASH\_BOARD理賠情報統計資料區取出, 逐筆進行各項資料合理性檢核作業
3. 執行主查詢1
   1. 執行[參考SQL1] 查詢條件參數如下
      1. DATA\_DATE\_STR = $ STR\_DATE
      2. DATA\_DATE\_END = $ END\_DATE
      3. 如果$STAT\_TYPE 不為空值, STAT\_TYPE = $STAT\_TYPE
      4. 如果$SYS\_NO 不為空值, SYS\_NO = $SYS\_NO
      5. 如果$CLAM\_DIV\_NO 不為空值, CLAM\_DIV\_NO = $CLAM\_DIV\_NO
   2. 若讀取異常(不含查無資料)，則設定：
      1. 訊息中文＝“主查詢讀取異常”，
      2. 摘　　要＝DBException，
      3. CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，
      4. 錯誤件數++，
      5. RollBack回程式初始狀態資料，結束程式且程式執行結果異常。
   3. 輸入件數 = 主查詢筆數
   4. 檢核暫存變數
      1. SET $不合規訊息\_LIST = 空的LIST物件
      2. SET $DIV\_SUM\_MAP = 空的MAP
   5. 執行檢核, 針對主查詢1的結果逐筆進行系列檢核, 每筆資料檢查完有錯誤再一次輸出
      1. SET $KEY值 = 主查詢.DATA\_DATE,主查詢.STAT\_TYPE,主查詢.SYS\_NO,主查詢.CLAM\_DIV\_NO
      2. SET $MAP\_KEY = 主查詢.DATA\_DATE,主查詢.STAT\_TYPE ,主查詢.CLAM\_DIV\_NO
      3. 不得為空\_顯示層, 邏輯勾稽\_顯示層 及邏輯勾稽\_資料\_ , 檢核規則及不合規訊息字串如下
         1. 有產生不合規訊息 就先塞入$不合規訊息\_LIST

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 項目 | 檢核規則 | 不合規訊息 |
|  | 理賠件數 | 如果主查詢.SYS\_NO = ‘A’ 且 ( 檢查主查詢.CNT為空值 或CNT = 0 ) 🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : A\_1  CHK\_LIST迴歸明細 : A\_1\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢.CNT |
|  | 理賠金額 | 如果主查詢.SYS\_NO = ‘A’ 且 (檢查主查詢.PAY\_AMT為空值 或 PAY\_AMT = 0) 🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : A\_2  CHK\_LIST迴歸明細 : A\_2\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. PAY\_AMT |
|  | 平均結案天數 | 1. 如果主查詢.SYS\_NO = ‘0’ 且 (檢查主查詢.PRCS\_DAY為空值 或 需 PRCS\_DAY = 0)    1. 如果統計週期是月年 或 (統計週期是日 且統日期 在 次月2號之後)       1. 🡪執行不合規訊息處理    2. 不然       1. 跳過不檢核 | 呼叫 **setChkList() BY參數**  TYPE類型 : A\_3  CHK\_LIST迴歸明細 : A\_3\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. PRCS\_DAY |
|  | 實質不給付件數 | 如果主查詢.SYS\_NO = ‘A’ 且 (檢查主查詢.NOPAY\_CNT為空值 或 NOPAY\_CNT = 0) 🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : A\_4  CHK\_LIST迴歸明細 : A\_4\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. NOPAY\_CNT |
|  | 實質不給付金額 | 如果主查詢.SYS\_NO = ‘A’ 且 (檢查 主查詢.NOPAY\_AMT為空值 且 NOPAY\_AMT = 0) 🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : A\_5  CHK\_LIST迴歸明細 : A\_5\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. NOPAY\_AMT |
|  | 業務別符合現有項目 | 主查詢. SYS\_NO NOT IN ( 0,A,G,H) 🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : B\_1  CHK\_LIST迴歸明細 : B\_1\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. SYS\_NO |
|  | 理賠單位符合現有項目 | 主查詢. CLAM\_DIV\_NO 不在 代碼中文AA/AAH5\_B332\_CLAM\_DIV設定的代碼值 之中🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_1  CHK\_LIST迴歸明細 : C\_1\_LIST  KEY\_DATA資料鍵值 : $KEY值  NUM\_DATA數字資料 : 主查詢. CLAM\_DIV\_NO |

* 1. 暫存各單位的各項數字
     1. SET $ONE\_MAP = $DIV\_SUM\_MAP.KEY 為 $MAP\_KEY 取回的MAP
     2. 如果 $ONE\_MAP 是 NULL
        1. SET $ONE\_MAP = 空的MAP
     3. 如果 主查詢.SYS\_NO = ‘0’, 塞入$ONE\_MAP 如下
        1. KEY = 0\_CNT , VALUE = 主查詢. CNT
        2. KEY = 0\_PAY\_AMT , VALUE = 主查詢.PAY\_AMT
        3. KEY = 0\_PRCS\_DAY , VALUE = 主查詢.PRCS\_DAY
        4. KEY = 0\_NOPAY\_CNT , VALUE = 主查詢.NOPAY\_CNT
        5. KEY = 0\_NOPAY\_AMT , VALUE = 主查詢.NOPAY\_AMT
     4. 如果 主查詢.SYS\_NO = ‘A’, 塞入$ONE\_MAP 如下
        1. KEY = A\_CNT , VALUE = 主查詢. CNT
        2. KEY = A\_PAY\_AMT , VALUE = 主查詢.PAY\_AMT
        3. KEY = A\_PRCS\_DAY , VALUE = 主查詢.PRCS\_DAY
        4. KEY = A\_NOPAY\_CNT , VALUE = 主查詢.NOPAY\_CNT
        5. KEY = A\_NOPAY\_AMT , VALUE = 主查詢.NOPAY\_AMT
     5. 如果 主查詢.SYS\_NO = ‘G’, 塞入$ONE\_MAP 如下
        1. KEY = G\_CNT , VALUE = 主查詢. CNT
        2. KEY = G\_PAY\_AMT , VALUE = 主查詢.PAY\_AMT
        3. KEY = G\_PRCS\_DAY , VALUE = 主查詢.PRCS\_DAY
        4. KEY = G\_NOPAY\_CNT , VALUE = 主查詢.NOPAY\_CNT
        5. KEY = G\_NOPAY\_AMT , VALUE = 主查詢.NOPAY\_AMT
     6. 如果 主查詢.SYS\_NO = ‘H’, 塞入$ONE\_MAP 如下
        1. KEY = H\_CNT , VALUE = 主查詢. CNT
        2. KEY = H\_PAY\_AMT , VALUE = 主查詢.PAY\_AMT
        3. KEY = H\_PRCS\_DAY , VALUE = 主查詢.PRCS\_DAY
        4. KEY = H\_NOPAY\_CNT , VALUE = 主查詢.NOPAY\_CNT
        5. KEY = H\_NOPAY\_AMT , VALUE = 主查詢.NOPAY\_AMT
     7. 將$ONE\_MAP 塞回$DIV\_SUM\_MAP.KEY
        1. KEY = $MAP\_KEY
        2. VALUE = $ONE\_MAP

1. 邏輯勾稽\_資料 檢核 (所有主查詢的資料都滾完了), 針對$DIV\_SUM\_MAP.KEY 內含逐筆key value 進行檢核
   1. SET $KEY\_STR = KEY值
   2. SET $DATA\_MAP = VALUE值

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 項目 | 檢核規則 | 不合規訊息 |
|  | 理賠件數 | 如果 $DATA\_MAP. A\_CNT != 0\_CNT + G\_CNT + H\_CNT🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_2  CHK\_LIST迴歸明細 : C\_2\_LIST  KEY\_DATA資料鍵值 : $KEY\_STR  NUM\_DATA數字資料 : A\_CNT, 0\_CNT ,G\_CNT, H\_CNT |
|  | 理賠金額 | 如果 $DATA\_MAP. A\_PAY\_AMT != 0\_PAY\_AMT + G\_PAY\_AMT + H\_PAY\_AMT🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_3  CHK\_LIST迴歸明細 : C\_3\_LIST  KEY\_DATA資料鍵值 : $KEY\_STR  NUM\_DATA數字資料 : A\_ PAY\_AMT, 0\_ PAY\_AMT ,G\_ PAY\_AMT, H\_ PAY\_AMT |
|  | 實質不給付件數 | 如果 $DATA\_MAP. A\_NOPAY\_CNT != 0\_NOPAY\_CNT + G\_NOPAY\_CNT + H\_NOPAY\_CNT🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_4  CHK\_LIST迴歸明細 : C\_4\_LIST  KEY\_DATA資料鍵值 : $KEY\_STR  NUM\_DATA數字資料 : A\_ NOPAY\_CNT, 0\_ NOPAY\_CNT ,G\_ NOPAY\_CNT, H\_ NOPAY\_CNT |
|  | 實質不給付金額 | 如果 $DATA\_MAP. A\_NOPAY\_AMT != 0\_NOPAY\_AMT + G\_NOPAY\_AMT + H\_NOPAY\_AMT🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_5  CHK\_LIST迴歸明細 : C\_5\_LIST  KEY\_DATA資料鍵值 : $KEY\_STR  NUM\_DATA數字資料 : A\_ NOPAY\_AMT, 0\_ NOPAY\_AMT ,G\_ NOPAY\_AMT, H\_ NOPAY\_AMT |
|  | 平均結案天數 | 如果 $DATA\_MAP. A\_PRCS\_DAY > 15🡪執行不合規訊息處理 | 呼叫 **setChkList() BY參數**  TYPE類型 : C\_6  CHK\_LIST迴歸明細 : C\_6\_LIST  KEY\_DATA資料鍵值 : $KEY\_STR  NUM\_DATA數字資料 : $DATA\_MAP.PRCS\_DAY |

1. 寫入巡檢檔案 及 批次計數
   1. 寫入巡檢件數
      1. 組成 DTZZM010說明如下,各檢核項目代號 都要組成一筆DTZZM010 再呼叫 ZZ\_M0Z020. doInsertDTZZM010() 寫入巡檢件數登入
         1. 組成DTZZM010 共同項目塞值如下
            1. SHUT\_DT : $WORK\_DATE
            2. SYS\_CODE : AA
            3. JOB\_NAME : JAAADH503
            4. PROGRAM : AAH5\_B332

|  |  |  |
| --- | --- | --- |
| 序號 | CNT\_ITEM | CNT\_VALUE |
|  | AAH50001 | A\_1\_LIST.筆數 - 1 |
|  | AAH50002 | A\_2\_LIST.筆數 - 1 |
|  | AAH50003 | A\_3\_LIST.筆數 - 1 |
|  | AAH50004 | A\_4\_LIST.筆數 - 1 |
|  | AAH50005 | A\_5\_LIST.筆數 - 1 |
|  | AAH50006 | B\_1\_LIST.筆數 - 1 |
|  | AAH50007 | C\_1\_LIST.筆數 - 1 |
|  | AAH50008 | C\_2\_LIST.筆數 - 1 |
|  | AAH50009 | C\_3\_LIST.筆數 - 1 |
|  | AAH50010 | C\_4\_LIST.筆數 - 1 |
|  | AAH50011 | C\_5\_LIST.筆數 - 1 |
|  | AAH50012 | C\_6\_LIST.筆數 - 1 |

* 1. 寫入驗證明細
     1. 針對上表各項CNT\_ITEM 都進行下列動作
        1. 若呼叫模組有誤, 處理如下
           1. Log.fatal（“寫入驗證明細: “＋$錯誤訊息），
           2. 訊息中文＝“傳入參數異常”，
           3. 摘　　要＝$錯誤訊息，
           4. CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，
           5. 錯誤件數++，
           6. RollBack回程式初始狀態資料，結束程式且程式執行結果異常。
        2. 清除當日資料 ,
           1. 呼叫 ZZ\_M4Z001.doClear() BY 參數

VRF\_NAME : $CNT\_ITEM

INPUT\_DATE : $WORK\_DATE

* + - 1. 寫入驗證資料
         1. 呼叫 ZZ\_M4Z001. doInsert() BY 參數如下, 寫入巡檢驗證明細

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序號 | VRF\_NAME | INPUT\_DATE | returnList | INPUT\_COUNT | INPUT\_AMT |
|  | AAH50001 | $WORK\_DATE | A\_1\_LIST | A\_1\_LIST.筆數 | 0 |
|  | AAH50002 | $WORK\_DATE | A\_2\_LIST | A\_2\_LIST.筆數 | 0 |
|  | AAH50003 | $WORK\_DATE | A\_3\_LIST | A\_3\_LIST.筆數 | 0 |
|  | AAH50004 | $WORK\_DATE | A\_4\_LIST | A\_4\_LIST.筆數 | 0 |
|  | AAH50005 | $WORK\_DATE | A\_5\_LIST | A\_5\_LIST.筆數 | 0 |
|  | AAH50006 | $WORK\_DATE | B\_1\_LIST | B\_1\_LIST.筆數 | 0 |
|  | AAH50007 | $WORK\_DATE | C\_1\_LIST | C\_1\_LIST.筆數 | 0 |
|  | AAH50008 | $WORK\_DATE | C\_2\_LIST | C\_2\_LIST.筆數 | 0 |
|  | AAH50009 | $WORK\_DATE | C\_3\_LIST | C\_3\_LIST.筆數 | 0 |
|  | AAH50010 | $WORK\_DATE | C\_4\_LIST | C\_4\_LIST.筆數 | 0 |
|  | AAH50011 | $WORK\_DATE | C\_5\_LIST | C\_5\_LIST.筆數 | 0 |
|  | AAH50012 | $WORK\_DATE | C\_6\_LIST | C\_6\_LIST.筆數 | 0 |

* 1. 寫入批次件數
     1. 設定件數
        1. A\_1錯誤件數 = A\_1\_LIST.筆數 - 1
        2. A\_2錯誤件數 = A\_2\_LIST.筆數 - 1
        3. A\_3錯誤件數 = A\_3\_LIST.筆數 - 1
        4. A\_4錯誤件數 = A\_4\_LIST.筆數 - 1
        5. A\_5錯誤件數 = A\_5\_LIST.筆數 - 1
        6. B\_1錯誤件數 = B\_1\_LIST.筆數 - 1
        7. C\_1錯誤件數 = C\_1\_LIST.筆數 - 1
        8. C\_2錯誤件數 = C\_2\_LIST.筆數 - 1
        9. C\_3錯誤件數 = C\_3\_LIST.筆數 - 1
        10. C\_4錯誤件數 = C\_4\_LIST.筆數 - 1
        11. C\_5錯誤件數 = C\_5\_LIST.筆數 - 1
        12. C\_6錯誤件數 = C\_6\_LIST.筆數 - 1
     2. CALL batch.CountManager(批次作業件數記錄模組)，記錄輸入件數，輸出件數及錯誤件數

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | | **setChkList** | | | |
| **功能說明** | |  | | | |
| **Javadoc** | | 傳入調查股代號， 回傳是否為調查股股長代理人, 如果是,回傳委託的股長id;不是就回傳空字串. 會加入 CACHE MAP | | | |
| **類型** | | □單檔維護　■查詢　□計算　□檢核 | | | □是否迴歸 |
| **輸入參數** | | | | | |
| **項次** | **參數名稱** | | **格式** | **說明(檢查規則)** | |
|  | TYPE類型 | | String |  | |
|  | CHK\_LIST迴歸明細 | | List |  | |
|  | KEY\_DATA資料鍵值 | | String |  | |
|  | NUM\_DATA數字資料 | | String |  | |
| **輸出參數(單筆)** | | | | | |
|  | 無 | |  |  | |

程式內容說明：

1. 起始：
2. 檢核：檢核完所有錯誤一次拋出, 各參數檢核及不合規訊息如下
   1. TYPE類型
      1. 檢核 : 不得為空值
      2. 不合規訊息 : 類型需有值
   2. CHK\_LIST迴歸明細
      1. 檢核 : 不得為NULL
      2. 不合規訊息 : 迴歸明細不得為NULL
   3. KEY\_DATA 資料鍵值
      1. 檢核: 不得為空值
      2. 不合規訊息 : 資料鍵值 需有值
   4. ~~NUM\_DATA數字資料~~
      1. ~~檢核: 不得為空值~~
      2. ~~不合規訊息: 數字資料需有值~~
3. 依照TYPE類型 對 資料鍵值 及 數字資料 進行拆解, 再塞入迴歸明細
   1. **如果**TYPE類型 = A\_1, A\_2, A\_3, A\_4, A\_5, A\_5,B\_1,C\_1
      1. **將**資料鍵值 依照 ‘,’ 分解, 取出依序為 $DATA\_DATE, $ STAT\_TYPE, $ SYS\_NO, $ CLAM\_DIV\_NO
      2. 1.6.1 SET $DETAIL\_LIST = 空的LIST
      3. 1.6.2 $DETAIL\_LIST 塞入 $DATA\_DATE
      4. 1.6.3 $DETAIL\_LIST 塞入 $ STAT\_TYPE
      5. 1.6.4 $DETAIL\_LIST 塞入 $ SYS\_NO
      6. 1.6.5 $DETAIL\_LIST 塞入 ‘$ CLAM\_DIV\_NO
      7. 1.6.8 $DETAIL\_LIST 塞入 NUM\_DATA數字資料
      8. 1.6.9 CHK\_LIST迴歸明細 塞入 $DETAIL\_LIST
   2. **如果**TYPE類型 = C\_2, C\_3, C\_4, C\_5
      1. **將**資料鍵值 依照 ‘,’ 分解, 取出依序為 $DATA\_DATE, $ STAT\_TYPE, $CLAM\_DIV\_NO
      2. **將**數字資料 依照 ‘,’ 分解, 取出依序為 $A\_NUM, $0\_NUM, $G\_NUM, $H\_NUM
      3. SET $DETAIL\_LIST = 空的LIST
      4. $DETAIL\_LIST 塞入 $DATA\_DATE
      5. $DETAIL\_LIST 塞入 $ STAT\_TYPE
      6. $DETAIL\_LIST 塞入 ‘$ CLAM\_DIV\_NO
      7. $DETAIL\_LIST 塞入 $A\_NUM
      8. $DETAIL\_LIST 塞入 $0\_NUM
      9. $DETAIL\_LIST 塞入 $G\_NUM
      10. $DETAIL\_LIST 塞入 $H\_NUM
   3. **如果**TYPE類型 = C\_6
      1. **將**資料鍵值 依照 ‘\_’ 分解, 取出依序為 $DATA\_DATE, $ STAT\_TYPE, $CLAM\_DIV\_NO
      2. 1.6.1 SET $DETAIL\_LIST = 空的LIST
      3. 1.6.2 $DETAIL\_LIST 塞入 $DATA\_DATE
      4. 1.6.3 $DETAIL\_LIST 塞入 $ STAT\_TYPE
      5. 1.6.5 $DETAIL\_LIST 塞入 ‘$ CLAM\_DIV\_NO
      6. 1.6.8 $DETAIL\_LIST 塞入 NUM\_DATA數字資料
      7. CHK\_LIST迴歸明細 塞入 $DETAIL\_LIST
4. RETURN

**[參考SQL1]**

**SELECT \***

**FROM DBGD.DTGDC009 GDC9**

**WHERE GDC9.DATA\_DATE BETWEEN ':DATA\_DATE\_STR' AND ':DATA\_DATE\_END'**

**[ AND GDC9.STAT\_TYPE = ':STAT\_TYPE' ]**

**[ AND GDC9.SYS\_NO = ':SYS\_NO' ]**

**[ AND GDC9.CLAM\_DIV\_NO = ':CLAM\_DIV\_NO' ]**

**WITH UR**

**~~[參考SQL2]~~**

~~WITH A\_DATA AS (~~

~~SELECT DATA\_DATE, STAT\_TYPE, SYS\_NO, CLAM\_DIV\_NO, CNT~~

~~FROM DBGD.DTGDC009 GDC9~~

~~WHERE GDC9.DATA\_DATE BETWEEN ':DATA\_DATE\_STR' AND ':DATA\_DATE\_END'~~

~~[ AND GDC9.STAT\_TYPE = ':STAT\_TYPE' ]~~

~~AND GDC9.SYS\_NO = 'A'~~

~~[ AND GDC9.CLAM\_DIV\_NO = ':CLAM\_DIV\_NO' ]~~

~~),~~

~~0\_DATA AS (~~

~~SELECT DATA\_DATE, STAT\_TYPE, SYS\_NO, CLAM\_DIV\_NO, CNT~~

~~FROM DBGD.DTGDC009 GDC9~~

~~WHERE GDC9.DATA\_DATE BETWEEN ':DATA\_DATE\_STR' AND ':DATA\_DATE\_END'~~

~~[ AND GDC9.STAT\_TYPE = ':STAT\_TYPE' ]~~

~~AND GDC9.SYS\_NO = '0'~~

~~[ AND GDC9.CLAM\_DIV\_NO = ':CLAM\_DIV\_NO' ]~~

~~),~~

~~G\_DATA AS (~~

~~SELECT DATA\_DATE, STAT\_TYPE, SYS\_NO, CLAM\_DIV\_NO, CNT~~

~~FROM DBGD.DTGDC009 GDC9~~

~~WHERE GDC9.DATA\_DATE BETWEEN ':DATA\_DATE\_STR' AND ':DATA\_DATE\_END'~~

~~[ AND GDC9.STAT\_TYPE = ':STAT\_TYPE' ]~~

~~AND GDC9.SYS\_NO = 'G'~~

~~[ AND GDC9.CLAM\_DIV\_NO = ':CLAM\_DIV\_NO' ]~~

~~),~~

~~H\_DATA AS (~~

~~SELECT DATA\_DATE, STAT\_TYPE, SYS\_NO, CLAM\_DIV\_NO, CNT~~

~~FROM DBGD.DTGDC009 GDC9~~

~~WHERE GDC9.DATA\_DATE BETWEEN ':DATA\_DATE\_STR' AND ':DATA\_DATE\_END'~~

~~[ AND GDC9.STAT\_TYPE = ':STAT\_TYPE' ]~~

~~AND GDC9.SYS\_NO = 'H'~~

~~[ AND GDC9.CLAM\_DIV\_NO = ':CLAM\_DIV\_NO' ]~~

~~)~~

~~SELECT~~

~~A\_DATA.DATA\_DATE,~~

~~A\_DATA.STAT\_TYPE,~~

~~A\_DATA.CLAM\_DIV\_NO,~~

~~A\_DATA.CNT A\_CNT,~~

~~VALUE(0\_DATA.CNT,~~**~~0~~**~~) 0\_CNT,~~

~~VALUE(G\_DATA.CNT,~~**~~0~~**~~) G\_CNT,~~

~~VALUE(H\_DATA.CNT,~~**~~0~~**~~) H\_CNT~~

~~FROM A\_DATA~~

~~LEFT JOIN 0\_DATA~~

~~ON A\_DATA.DATA\_DATE = 0\_DATA.DATA\_DATE AND A\_DATA.STAT\_TYPE = 0\_DATA.STAT\_TYPE AND A\_DATA.CLAM\_DIV\_NO = 0\_DATA.CLAM\_DIV\_NO~~

~~LEFT JOIN G\_DATA~~

~~ON A\_DATA.DATA\_DATE = G\_DATA.DATA\_DATE AND A\_DATA.STAT\_TYPE = G\_DATA.STAT\_TYPE AND A\_DATA.CLAM\_DIV\_NO = G\_DATA.CLAM\_DIV\_NO~~

~~LEFT JOIN H\_DATA~~

~~ON A\_DATA.DATA\_DATE = H\_DATA.DATA\_DATE AND A\_DATA.STAT\_TYPE = H\_DATA.STAT\_TYPE AND A\_DATA.CLAM\_DIV\_NO = H\_DATA.CLAM\_DIV\_NO~~

~~WITH UR~~