|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改日期 | 版本 | 修改原因 | 修改人姓名 | 立案單號 |
| 2020/03/03 | 1 | Created  電腦作業申請書191210000755\_CRSS2.0+KN導入 | 楊智偉 | 191216000725 |

**一、程式功能概述**

|  |  |  |
| --- | --- | --- |
| 程式功能 | CRSS2\_保費資料\_墊繳資料抓取 | |
| 程式名稱 | AAV4\_B010.java | |
| 作業方式 | BATCH | |
| 概要說明 | CRSS2\_保費資料\_墊繳資料抓取 | |
| 需求單位 | 商業智能科 | |
| 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. 1. | 契約墊繳記錄 | DTACQ010 | ■ | □ | □ | □ |
|  | 契約墊繳歷史記錄 | DTACQ015 | ■ | □ | □ | □ |
|  | 契約墊繳餘額明細檔 | DTACQ070 | ■ | □ | □ | □ |
|  | CRSS2\_保費資料\_墊繳停效 | DTAAV430 | □ | ■ | ■ | □ |

**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 | CRSS2.0+KN導入 |
| 業務別 | AA |
| 次系統名稱 | V4 |
| 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 |
|  | POLICY\_NO保單號碼 | 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. Set $WORK\_TIMESTAMP = 系統TIMESTAMP
   3. 取得系統SHUTDOWN DAY： CALL CathayDate模組取得系統SHUTDOWN DAY：
   4. 件數歸零：
      1. START、
      2. 輸入件數、
      3. 錯誤件數、
      4. 更新件數、
      5. 新增件數
   5. 查詢條件變數 : 都先給null 初始化
      1. $STR\_DATE 開始日期
      2. $POLICY\_NO保單號碼
   6. 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. 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 >=1
         1. $STR\_DATE = 傳入. 開始日期，
      3. 如果傳入參數個數 >=2
         1. $STR\_DATE = 傳入. 開始日期，
         2. $POLICY\_NO = 傳入.保單號碼，
   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：
         1. 須為西元年格式, 合理日期
         2. 不符時, $錯誤訊息 加上 : 開始日期須為西元年格式.
      3. 如果$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 : 將指定日期區間之後 (其他選擇性條件為保單號碼), 符合條件的保費墊繳資料取出, 逐筆寫出
3. 執行主查詢1
   1. 執行[參考SQL1] 查詢條件參數如下
      1. 保費應繳日期PAY\_DATE\_POL = $ STR\_DATE
      2. 如果$POLICY\_NO 不為空值, POLICY\_NO = $ POLICY\_NO
   2. 若查無資料，則設定：
      1. 訊息中文＝“主查詢查無資料”，
      2. 摘　　要＝ “開始日期”+ $STR\_DATE
      3. CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，
      4. 正常結束程式。
   3. 若讀取常(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。
   4. 輸入件數 = 主查詢筆數
   5. 逐筆輸出
      1. 寫入DTAAV430 (CRSS2\_保費資料\_墊繳停效 ) 如下
         1. POLICY\_NO : 批次查詢. POLICY\_NO
         2. DATA\_TYPE : 1 //自動墊繳
         3. MAIN\_RD\_CODE : 0 //自動墊繳都給0
         4. DATA\_DATE : 批次查詢. PAY\_DATE\_POL
         5. INPUT\_TIME : $WORK\_TIMESTAMP
      2. 寫入成功 : 新增件數++
      3. 寫入失敗 :
         1. 如果是資料重複, 改用UPDATE
            1. 更新 DTAAV430

SET 欄位

DATA\_DATE : 批次查詢. PAY\_DATE\_POL

INPUT\_TIME : $WORK\_TIMESTAMP

條件

POLICY\_NO : 批次查詢. POLICY\_NO

DATA\_TYPE : 1 //自動墊繳

MAIN\_RD\_CODE : 0 //自動墊繳都給0

* + - * 1. 更新成功

更新件數++

* + - * 1. 不然

錯誤件數++

寫入ErrorLog, 處理如下

Log.fatal（“更新DTAAV430: “＋$錯誤訊息），

訊息中文＝“更新DTAAV430異常”，

摘　　要＝$錯誤訊息，

CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，

* + - 1. 不然
         1. 錯誤件數++
         2. 寫入ErrorLog, 處理如下

Log.fatal（“寫入DTAAV430: “＋$錯誤訊息），

訊息中文＝“DTAAV430異常”，

摘　　要＝$錯誤訊息，

CALL batch.ErrorLog (異常訊息記錄模組)，記錄錯誤訊息，

* 1. 寫入批次件數
     1. 設定件數
        1. 輸入件數
        2. 錯誤件數
        3. 更新件數
        4. 新增件數
     2. CALL batch.CountManager(批次作業件數記錄模組)，記錄所有批次件數

**[參考SQL1]**

SELECT POLICY\_NO, PAY\_DATE\_POL FROM DBAC.DTACQ010 ACQ010

WHERE PAY\_DATE\_POL >= ':PAY\_DATE\_POL'

[ AND POLICY\_NO = ':POLICY\_NO' ]

UNION

SELECT POLICY\_NO, PAY\_DATE\_POL FROM DBAC.DTACQ015 ACQ015

WHERE PAY\_DATE\_POL >= ':PAY\_DATE\_POL'

[ AND POLICY\_NO = ':POLICY\_NO' ]

UNION

SELECT DISTINCT POLICY\_NO, PAY\_DATE\_POL FROM DBAC.DTACQ070 ACQ070

WHERE PAY\_DATE\_POL >= ':PAY\_DATE\_POL'

[ AND POLICY\_NO = ':POLICY\_NO' ]

WITH UR