

# 核心小程序整理

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 部门 | 日期 | 备注 |
| V1.0 | 邓县华 | 核心组 | 2015/2/6 | 创建文档 |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

[核心小程序整理 1](#_Toc415819631)

[目录 3](#_Toc415819632)

**[1，](#_Toc415819633)****[核心补支付单状态程序PayTimeoutJob](#_Toc415819633)** [5](#_Toc415819633)

[1.1 程序流程图： 5](#_Toc415819634)

[1.2 程序中用到的sql： 7](#_Toc415819635)

**[2，](#_Toc415819636)****[核心补业务单状态程序Fixstat](#_Toc415819636)** [8](#_Toc415819636)

[2.1 存储过程如下： 8](#_Toc415819637)

[2.2 程序流程图： 9](#_Toc415819638)

[2.3 程序中用到的sql： 10](#_Toc415819639)

**[3，](#_Toc415819640)****[自动转账程序AutoTransfer](#_Toc415819640)** [11](#_Toc415819640)

[3.1 存储过程如下： 11](#_Toc415819641)

[3.2 程序流程图： 12](#_Toc415819642)

[3.3 程序中用到的sql： 12](#_Toc415819643)

**[4，](#_Toc415819644)****[自动提现程序AutoWithdrawal](#_Toc415819644)** [13](#_Toc415819644)

[4.1 存储过程如下： 13](#_Toc415819645)

[4.2 程序流程图： 13](#_Toc415819646)

[4.3 程序中用到的sql： 14](#_Toc415819647)

**[5，](#_Toc415819648)****[核心补业务单的RET\_ID程序FixRetId](#_Toc415819648)** [15](#_Toc415819648)

[5.1 程序流程图： 15](#_Toc415819649)

[5.2 程序中用到的sql： 16](#_Toc415819650)

**[6，](#_Toc415819651)****[处理清结算对出的差异单程序CleaningStateSync](#_Toc415819651)** [17](#_Toc415819651)

[6.1 程序流程图： 17](#_Toc415819652)

[6.2 程序中用到的sql： 19](#_Toc415819653)

**[7，](#_Toc415819654)****[待结算程序schemeCarryForward](#_Toc415819654)** [19](#_Toc415819654)

[7.1 程序流程图： 19](#_Toc415819655)

[7.2 程序中用到的sql： 21](#_Toc415819656)

**[8，](#_Toc415819657)****[延迟提现程序withdrawDelayed](#_Toc415819657)** [24](#_Toc415819657)

[8.1 程序流程图： 24](#_Toc415819658)

[8.2 程序中用到的sql： 25](#_Toc415819659)

**[9，](#_Toc415819660)****[酬金返还程序AutoPayReward](#_Toc415819660)** [27](#_Toc415819660)

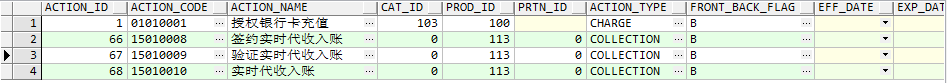
[9.1 程序流程图： 27](#_Toc415819661)

[9.2 程序中用到的sql： 29](#_Toc415819662)

核心小程序都在工程的job文件目录下。

1. **核心补支付单状态程序PayTimeoutJob**

目录job/AuthorizedCardRcg,编译出来的程序名：PayTimeoutJob，每5分钟执行一次。配置文件：config/PayTimeoutJob.ini,该程序只处理下面这4种业务。



1.1 程序流程图：



1.2 程序中用到的sql：

获取支付超时的订单数据：

SELECT O.ORDER\_ID

FROM T\_SCS\_ORDER O, T\_SCS\_ACTLIST A

WHERE TO\_CHAR(O.ACCT\_DATE, 'YYYYMMDD') = TO\_CHAR(SYSDATE, 'YYYYMMDD')

AND O.PAY\_STAT = 'S0E'

AND A.ORDER\_ID = O.ORDER\_ID

AND A.ACTION\_ID IN (1, 66, 67, 68);

SELECT O.ORDER\_ID,

TO\_CHAR(O.ACCT\_DATE, 'YYYYMMDDHH24MISS') AS STR\_ACCT\_DATE,

O.STAT,

O.PAY\_STAT,

O.ACT\_STAT,

O.CHANNEL\_TYPE,

O.AREA\_CODE,

A.ACTION\_ID,

P.PAY\_TYPE,

P.PAY\_ORG,

P.PGW\_PAY\_ORG,

P.ACCT\_CODE,

P.EVENT\_SEQ

FROM T\_SCS\_ORDER O, T\_SCS\_ACTLIST A, T\_SCS\_PAYMENT P

WHERE A.ORDER\_ID = O.ORDER\_ID

AND P.ORDER\_ID = O.ORDER\_ID

AND O.ORDER\_ID = '%s';

SELECT AR.VALUE1 AS BANKCARDID

FROM T\_SCS\_ACT\_ATTR AR, T\_SCS\_ACTLIST A

WHERE A.ORDER\_ID = '%s'

AND AR.ACTLIST\_ID = A.ACTLIST\_ID

AND AR.ATTR\_ID = 1048;

SELECT AR.VALUE1 AS BANKCARDID

FROM T\_SCS\_ACT\_ATTR AR, T\_SCS\_ACTLIST A

WHERE A.ORDER\_ID = '%s'

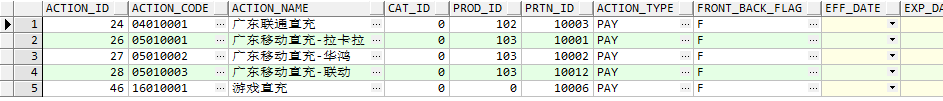
AND AR.ACTLIST\_ID = A.ACTLIST\_ID

AND AR.ATTR\_ID = 1049;

SELECT DEAL\_TIMES,STAT FROM T\_SCS\_WRONG\_ORDER WHERE ORDER\_ID = '%s';

1. **核心补业务单状态程序Fixstat**

目录job/ScanTime,编译出来的程序名：Fixstat，每2分钟执行一次。配置文件：config/job.ini,该程序只处理下面这5种业务。



首先有一个存储过程会定时往表T\_SCS\_WRONG\_ORDER中插入程序所需的数据。然后程序关联这张表得到所需的数据做后续处理。

2.1 存储过程如下：

CREATE OR REPLACE PROCEDURE P\_SCS\_WRONG\_ORDER\_COLLECT AS

CURSOR C\_ORDERID IS

SELECT ORD.ORDER\_ID,

PAY.EVENT\_SEQ PAY\_EVENT\_SEQ,

ACT.EVENT\_SEQ ACT\_EVENT\_SEQ,

ORD.ACCT\_DATE ACCT\_DATE,

ORD.STAT ORD\_STAT,

PAY.STAT PAY\_STAT,

ACT.STAT ACT\_STAT,

ACT.ACTION\_ID

FROM T\_SCS\_ORDER ORD, T\_SCS\_PAYMENT PAY, T\_SCS\_ACTLIST ACT

WHERE ORD.ORDER\_ID = PAY.ORDER\_ID

AND ORD.ORDER\_ID = ACT.ORDER\_ID

AND ACT.ACTION\_ID IN (24, 26, 27, 28, 46,82,83,84)----这里应该有问题？

AND (ACT.RET\_CODE = '0031' OR ACT.RET\_CODE = '0034' OR ACT.RET\_CODE = '9090')

AND PAY.RET\_CODE = '0000'

AND PAY.STAT = 'S0S'

AND ACT.STAT = 'S0E'

AND (ORD.STAT = 'S0C' OR ORD.STAT = 'S0D')

AND TO\_CHAR(ORD.ACCT\_DATE, 'YYYYMMDD') =

TO\_CHAR(SYSDATE, 'YYYYMMDD')

AND ORD.ORDER\_ID NOT IN

(SELECT WR.ORDER\_ID FROM T\_SCS\_WRONG\_ORDER WR);

BEGIN

FOR REC IN C\_ORDERID LOOP

DBMS\_OUTPUT.PUT\_LINE(REC.ORDER\_ID);

INSERT INTO T\_SCS\_WRONG\_ORDER

(ORDER\_ID,

PAY\_EVENT\_SEQ,

ACT\_EVENT\_SEQ,

ACCT\_DATE,

DEAL\_DATE,

ORD\_STAT,

PAY\_STAT,

ACT\_STAT,

PAY\_RETCODE,

PAY\_MSG,

ACT\_RETCODE,

ACT\_MSG,

DEAL\_TIMES,

ACTION\_ID,

STAT,

CREATE\_DATE)

VALUES

(REC.ORDER\_ID,

REC.PAY\_EVENT\_SEQ,

REC.ACT\_EVENT\_SEQ,

REC.ACCT\_DATE,

'',

REC.ORD\_STAT,

REC.PAY\_STAT,

REC.ACT\_STAT,

'',

'',

'',

'',

0,

REC.ACTION\_ID,

'S0A',

SYSDATE);

END LOOP;

COMMIT;

END P\_SCS\_WRONG\_ORDER\_COLLECT;

2.2 程序流程图：



2.3 程序中用到的sql：

SELECT ORD.ORDER\_ID,

ORD.ORDER\_TYPE,

TO\_CHAR(ORD.ACCT\_DATE, 'YYYYMMDDHH24MISS') AS ACCT\_DATE,

PDM.ACTION\_CODE,

ORD.CHANNEL\_TYPE,

PAY.PAY\_TYPE,

ORD.AREA\_CODE,

ACT.EVENT\_SEQ,

PROD.PROD\_CODE,

ORD.TERM\_ID,

ORD.CUST\_ID,

WR.DEAL\_TIMES

FROM T\_SCS\_ORDER ORD,

T\_PDM\_ACTION PDM,

T\_SCS\_PAYMENT PAY,

T\_SCS\_ACTLIST ACT,

T\_PDM\_PRODUCT PROD,

T\_SCS\_WRONG\_ORDER WR

WHERE ORD.ORDER\_ID = ACT.ORDER\_ID

AND ORD.ORDER\_ID = WR.ORDER\_ID

AND ACT.ORDER\_ID = WR.ORDER\_ID

AND ORD.ORDER\_ID = PAY.ORDER\_ID

AND PROD.PROD\_ID = ACT.PROD\_ID

AND (ORD.STAT = 'S0C' OR ORD.STAT = 'S0D')

AND WR.STAT = 'S0A'

AND ACT.ACTION\_ID IN

(SELECT ACTION\_ID

FROM T\_PDM\_ACTION

WHERE ACTION\_CODE IN

('04010001', '05010001', '05010002', '05010003', '16010001'))

AND PDM.ACTION\_ID = ACT.ACTION\_ID

AND ORD.ACT\_STAT = 'S0E'

AND (ACT.RET\_CODE = '0031' or ACT.RET\_CODE = '0034' OR

act.ret\_code = '9090')

AND ACT.STAT = 'S0E'

AND ORD.ORDER\_TYPE = 'OT001'

AND TO\_CHAR(ORD.ACCT\_DATE, 'YYYYMMDD') = TO\_CHAR(SYSDATE, 'YYYYMMDD');

1. **自动转账程序AutoTransfer**

目录job/Transfer,编译出来的程序名：AutoTransfer，每60秒执行一次。配置文件：config/job.ini。

首先有一个存储过程会定时往表t\_scs\_speoper\_list中插入程序所需的数据。然后程序关联这张表得到所需的数据做后续处理。

3.1 存储过程如下：



3.2 程序流程图：



3.3 程序中用到的sql：

SELECT SP.SEQ\_ID,SP.CUST\_CODE,SP.AREA\_CODE,SP.PROD\_CODE,SP.ACTION\_CODE,SP.PAY\_TYPE,SP.PAY\_ORG,SP.ACCT\_CODE,SP.OBJ\_CODE,SP.PAY\_MONEY,SP.FAIL\_TIMES,SP.CUST\_ID,AC.ACCT\_CODE AS OBJ\_ACCT\_CODE FROM T\_SCS\_SPEOPER\_LIST SP,T\_CUM\_ACCT AC WHERE SP.SPEOPER\_TYPE = '0006' and SP.FAIL\_TIMES<2 and SP.STAT in('S0A','S0W') AND AC.ACCT\_TYPE = 'ACCT002' AND AC.STAT = 'S0A' AND EXISTS (SELECT INFO.CUST\_ID FROM T\_CUM\_INFO INFO WHERE INFO.CUST\_CODE = SP.OBJ\_CODE and INFO.STAT = 'S0A' AND INFO.CUST\_ID = AC.CUST\_ID)

1. **自动提现程序AutoWithdrawal**

目录job/AutoCarry,编译出来的程序名：AutoWithdrawal，每60秒执行一次。配置文件：config/job.ini。

首先有一个存储过程会定时往表t\_scs\_speoper\_list中插入程序所需的数据。然后程序关联这张表得到所需的数据做后续处理。

4.1 存储过程如下：



4.2 程序流程图：



4.3 程序中用到的sql：

SELECT \* FROM (SELECT SP.SEQ\_ID,SP.AREA\_CODE,SP.CUST\_ID,SP.CUST\_CODE,SP.PROD\_ID,SP.PROD\_CODE,SP.ACTION\_ID,SP.ACTION\_CODE, SP.PAY\_TYPE,SP.PAY\_ORG,SP.ACCT\_CODE,SP.PAY\_MONEY, SP.OBJ\_CODE,SP.FAIL\_TIMES,

(SELECT ACCT\_CODE FROM T\_CUM\_ACCT WHERE ACCT\_TYPE = 'ACCT001' AND CUST\_ID = (SELECT CUST\_ID FROM T\_CUM\_INFO WHERE CUST\_CODE = SP.OBJ\_CODE AND STAT = 'S0A') AND STAT = 'S0A') AS OBJ\_ACCT\_CODE,

(SELECT BANK\_ID FROM T\_CUM\_ACCT WHERE ACCT\_TYPE = 'ACCT001' AND CUST\_ID = (SELECT CUST\_ID FROM T\_CUM\_INFO WHERE CUST\_CODE = SP.OBJ\_CODE AND STAT = 'S0A') AND STAT = 'S0A') AS BANK\_ID,

(SELECT ACCT\_NAME FROM T\_CUM\_ACCT WHERE ACCT\_TYPE = 'ACCT001' AND CUST\_ID = (SELECT CUST\_ID FROM T\_CUM\_INFO WHERE CUST\_CODE = SP.OBJ\_CODE AND STAT = 'S0A') AND STAT = 'S0A') AS OBJ\_ACCT\_NAME,

(SELECT ACCT\_ID FROM T\_CUM\_ACCT WHERE ACCT\_TYPE = 'ACCT001' AND CUST\_ID = (SELECT CUST\_ID FROM T\_CUM\_INFO WHERE CUST\_CODE = SP.OBJ\_CODE AND STAT = 'S0A') AND STAT = 'S0A') AS OBJ\_ACCT\_ID,

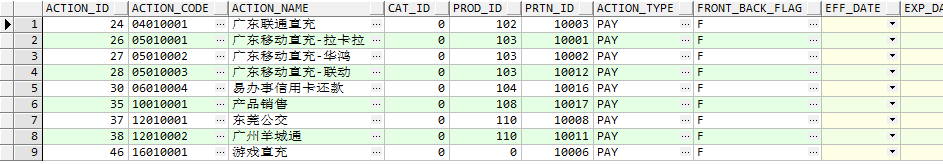
(SELECT VALUE1 FROM T\_CUM\_ACCT\_ATTR WHERE ACCT\_ID IN (SELECT ACCT\_ID FROM T\_CUM\_ACCT WHERE ACCT\_TYPE = 'ACCT001' AND CUST\_ID = (SELECT CUST\_ID FROM T\_CUM\_INFO

WHERE CUST\_CODE = SP.OBJ\_CODE AND STAT = 'S0A') AND STAT = 'S0A') AND ATTR\_ID = 2810 AND STAT = 'S0A') AS SIGN\_ID

FROM T\_SCS\_SPEOPER\_LIST SP WHERE SP.SPEOPER\_TYPE = '0007' AND SP.STAT = 'S0A' AND SP.FAIL\_TIMES < 2 ) WHERE ROWNUM <= 30

1. **核心补业务单的RET\_ID程序FixRetId**

目录job/FixRetId,编译出来的程序名：FixRetId，每天24点执行。配置文件：config/job.ini,该程序**不处理**下面这9种业务。



5.1 程序流程图：



5.2 程序中用到的sql：

获取业务单RET\_ID为空的数据：

SELECT ORD.ORDER\_ID,ORD.ORDER\_TYPE,TO\_CHAR(ORD.ACCT\_DATE,'YYYYMMDDHH24MISS') AS ACCT\_DATE,PDM.ACTION\_CODE,ORD.CHANNEL\_TYPE,

ORD.AREA\_CODE,ACT.EVENT\_SEQ,PROD.PROD\_CODE,ORD.TERM\_ID,ORD.CUST\_ID

FROM T\_SCS\_ORDER ORD,T\_PDM\_ACTION PDM,T\_SCS\_ACTLIST ACT, T\_PDM\_PRODUCT PROD

WHERE ORD.ORDER\_ID = ACT.ORDER\_ID AND PDM.ACTION\_TYPE = 'PAY' AND PROD.PROD\_ID = ACT.PROD\_ID

AND ACT.ACTION\_ID IN (SELECT ACTION\_ID FROM T\_PDM\_ACTION WHERE ACTION\_CODE NOT IN

('04010001','05010001','05010002','05010003','16010001','06010004','10010001','12010001','12010002'))

AND PDM.ACTION\_ID = ACT.ACTION\_ID

AND ACT.RET\_ID IS NULL AND ORD.ORDER\_TYPE = 'OT001' AND TO\_CHAR(ORD.ACCT\_DATE,'YYYYMMDD') = TO\_CHAR(SYSDATE-1,'YYYYMMDD');

1. **处理清结算对出的差异单程序CleaningStateSync**

目录job/UpdateOrderState,编译出来的程序名：CleaningStateSync，每天6点执行。配置文件：config/job.ini,该程序扫描清结算的文件做处理。

文件名为tz\_20150305\_gz.txt这种，20150305为当前时间，文件中的记录如下：

150301156722155,20150301,S0S,S0S,S0C,20150303140326,H03,补销账  
150301156772383,20150301,S0R,S0R,S0R,20150303140324,H01,退费

清结算的文件是程序启动脚本ftp到指定地址获取到的，程序运行时需要处理文件名称和批次日期（例如20131010）两个必要输入参数

6.1 程序流程图：



6.2 程序中用到的sql：

1. **待结算程序schemeCarryForward**

目录job/schemeCarryForward,编译出来的程序名： schemeCarryForward，每10分钟执行一次。配置文件：config/job.ini,该程序

7.1 程序流程图：



7.2 程序中用到的sql：

获取待结算结转数据sql：

SELECT DAY\_INFO FROM T\_SCS\_HOLIDAYLIST WHERE YEAR\_ID = 2015;

select a.\*,a.rowid from T\_PDM\_SCHEME a;

*---------.进行客户挂靠方案查找*

select a.\*,a.rowid from T\_SCS\_HOLIDAYLIST a;

SELECT \* FROM ( SELECT SCHEME\_ID,STAT,TO\_CHAR(EFF\_DATE,'YYYYMMDDHH24MISS')

AS EFF\_DATE, MAIN\_ID FROM T\_PDM\_SCHEME where

MAIN\_TYPE = 'CUST' and stat in ( 'S0A' ,'S0F')AND SYSDATE >= EFF\_DATE

ORDER BY ATTACHED\_ID ASC) WHERE ROWNUM >= 0;

select a.\*,a.rowid from T\_PDM\_SCHEME a where

MAIN\_TYPE = 'CUST' and stat in ( 'S0A' ,'S0F')AND SYSDATE >= EFF\_DATE

ORDER BY ATTACHED\_ID ASC ;

*--------1.挂靠关系中查找方案ID，先按商户查找*

SELECT SCHEME\_ID, ATTACHED\_ID

FROM T\_PDM\_SCHEME

WHERE MAIN\_ID ='60172313@ipos.com'

AND SYSDATE >= EFF\_DATE

AND STAT in ('S0A', 'S0F')

AND MAIN\_TYPE = 'CUST';

*-----2.方案基础表中确认有记录*

SELECT NAME,CODE,IS\_MULTI,POINT\_VALUE FROM T\_PDM\_SCHEMELIST WHERE SCHEME\_ID = '685' AND STAT = 'S0A';

*-----3.方案详情表中确认有记录*

SELECT SCHEME\_ID, ACTION\_ID，SETTLEMENT\_TYPE，SETTLEMENT\_PERIOD, WORK\_FLAG

FROM T\_PDM\_SETTLEMENT\_RULE

WHERE SCHEME\_ID = '685'

AND STAT = 'S0A'

AND SETTLEMENT\_TYPE IN ('ST103', 'ST102');

select a.\*,a.rowid from t\_sym\_dict a where a.dict\_id='ST102' and a.stat='S0A';

select a.\*,a.rowid from t\_sym\_dict a where a.dict\_typeid='SETTLEMENT\_TYPE' and a.stat='S0A';

select a.\*,a.rowid from t\_cum\_info a where a.cust\_code='60172313@ipos.com';

select a.\*,a.rowid from t\_cum\_acct a where a.cust\_id='241794';

select a.\*,a.rowid from t\_ebk\_deallog a where a.caac\_code='110441801000001026' and a.record\_date>sysdate-4;

*---机构查询*

SELECT MAIN\_ID,SCHEME\_ID,STAT,TO\_CHAR(EFF\_DATE,'YYYYMMDDHH24MISS')

AS EFF\_DATE,ATTACHED\_ID FROM T\_PDM\_SCHEME WHERE MAIN\_TYPE = 'ORG' AND STAT in ( 'S0A','S0F') AND SYSDATE >= EFF\_DATE ;

SELECT SCHEME\_ID, ATTACHED\_ID

FROM T\_PDM\_SCHEME

WHERE MAIN\_ID ='635297'

AND SYSDATE >= EFF\_DATE

AND STAT in ('S0A', 'S0F')

AND MAIN\_TYPE = 'ORG';

SELECT NAME,CODE,IS\_MULTI,POINT\_VALUE FROM T\_PDM\_SCHEMELIST WHERE SCHEME\_ID = '684' AND STAT = 'S0A';

SELECT SCHEME\_ID, ACTION\_ID，SETTLEMENT\_TYPE，SETTLEMENT\_PERIOD, WORK\_FLAG

FROM T\_PDM\_SETTLEMENT\_RULE

WHERE SCHEME\_ID = '684'

AND STAT = 'S0A'

AND SETTLEMENT\_TYPE IN ('ST103', 'ST102');

SELECT CUST\_CODE FROM T\_CUM\_INFO WHERE PRTN\_ID IN

(

SELECT PRTN\_ID FROM

(SELECT PRTN\_ID,PRTN\_TYPE FROM T\_PNM\_PARTNER START WITH PARENT\_ID = 635297

CONNECT BY PRIOR PRTN\_ID = PARENT\_ID )

WHERE PRTN\_TYPE in( 'PT901','PT404','PT401','PT301')

);

SELECT SCHEME\_ID, ACTION\_ID，SETTLEMENT\_TYPE，SETTLEMENT\_PERIOD, WORK\_FLAG

FROM T\_PDM\_SETTLEMENT\_RULE

WHERE SCHEME\_ID = '684'

AND STAT = 'S0A'

AND SETTLEMENT\_TYPE='ST102';

select trunc(sysdate) - trunc(sysdate,'YYYY') + 1 as DAY\_OVER from dual;

select TO\_CHAR(sysdate-3,'YYYYMMDD') AS NUMDATE from dual;

select a.\*,a.rowid from T\_EBK\_DEALLOG a;

*--ST103 按周期方式结转*

SELECT A.ORDER\_CODE,A.DEAL\_AMOUNT,A.CAAC\_CODE FROM T\_EBK\_DEALLOG A,T\_SCS\_ACTLIST B, T\_CUM\_INFO CUM,

T\_CUM\_ACCT ACCT,T\_EBK\_ACCTINFO INFO WHERE CUM.CUST\_ID = :CUST\_ID

AND CUM.CUST\_ID = ACCT.CUST\_ID AND ACCT.ACCT\_CODE = INFO.CAAC\_CODE AND

ACCT.ACCT\_TYPE = 'ACCT002' AND INFO.ACCT\_TYPE = '0300' AND

INFO.ACCT\_CODE = A.ACCT\_CODE

AND A.CAAC\_CODE != '999999999999999999' AND A.ORDER\_CODE = B.ORDER\_ID

AND B.STAT = 'S0S'

AND ROUND(TO\_NUMBER(sysdate - A.RECORD\_DATE) \* 24\*60) >=:settlement\_period\_string

if (action\_id\_info == "63") //如果为二维码支付

{

sqlAction = " AND B.ACTION\_ID in (135,:ACTION\_ID) ";

}

else

sqlAction = " AND B.ACTION\_ID = :ACTION\_ID ";

AND A.LOAN\_FLAG = '+' AND A.SCHEME\_STAT IS NULL

*--ST102按帐户结转*

SELECT ACTION\_ID FROM T\_PDM\_SCHEME\_ACTION\_RELA WHERE SETLEMENT = 'S0A';

SELECT A.ORDER\_CODE,A.DEAL\_AMOUNT,A.CAAC\_CODE FROM T\_EBK\_DEALLOG A,T\_SCS\_ACTLIST B, T\_CUM\_INFO CUM,

T\_CUM\_ACCT ACCT,T\_EBK\_ACCTINFO INFO WHERE CUM.CUST\_ID = :CUST\_ID

AND CUM.CUST\_ID = ACCT.CUST\_ID AND ACCT.ACCT\_CODE = INFO.CAAC\_CODE AND

ACCT.ACCT\_TYPE = 'ACCT002' AND INFO.ACCT\_TYPE = '0300' AND

INFO.ACCT\_CODE = A.ACCT\_CODE

AND A.CAAC\_CODE != '999999999999999999' AND A.ORDER\_CODE = B.ORDER\_ID

AND B.STAT = 'S0S' AND B.ACTION\_ID IN () AND A.LOAN\_FLAG = '+' AND A.SCHEME\_STAT IS NULL;

*--自动补充值*

SELECT A.ORDER\_CODE,A.DEAL\_AMOUNT,A.CAAC\_CODE FROM T\_EBK\_DEALLOG A,T\_SCS\_ACTLIST B, T\_CUM\_INFO CUM,

T\_CUM\_ACCT ACCT,T\_EBK\_ACCTINFO INFO WHERE CUM.CUST\_ID = :CUST\_ID

AND CUM.CUST\_ID = ACCT.CUST\_ID AND ACCT.ACCT\_CODE = INFO.CAAC\_CODE AND

ACCT.ACCT\_TYPE = 'ACCT002' AND INFO.ACCT\_TYPE = '0300' AND

INFO.ACCT\_CODE = A.ACCT\_CODE

AND A.CAAC\_CODE != '999999999999999999' AND A.ORDER\_CODE = B.ORDER\_ID

AND B.STAT = 'S0S' AND B.ACTION\_ID = '136' AND A.LOAN\_FLAG = '+' AND A.SCHEME\_STAT IS NULL;

1. **延迟提现程序withdrawDelayed**

目录job/withdrawDelayed,编译出来的程序名： withdrawDelayed，每2小时执行一次。配置文件：config/job.ini,该程序

8.1 程序流程图：



8.2 程序中用到的sql：

获取符合延迟提现的数据：

SELECT \* FROM ( SELECT ORDER\_ID,STAT,DELAY\_HOUR,TO\_CHAR(OPER\_TIME,'YYYYMMDDHH24MISS')

AS OPER\_TIME,RETRY\_COUNT,ORDER\_TPYE,CHANNEL\_TPYE,DEAL\_DATE,PRTN\_ID,CUST\_CODE,

RET\_CODE,RET\_INFOR,WORK\_FLAG FROM T\_SCS\_WITHDRAWAL where

ROUND(TO\_NUMBER(sysdate - oper\_time) \* 24 \* 60) >= delay\_hour and retry\_count > 0 and

retry\_count <= 3 and stat != 'S0C' and delay\_hour > 0

ORDER BY OPER\_TIME ASC) WHERE ROWNUM >= 0;

延迟提现回调用到的sql：

SELECT A.RULE\_TYPE

FROM T\_SCS\_WITHDRAWAL\_RULE A, T\_PNM\_PARTNER B, T\_CUM\_INFO C

WHERE A.STATUS IN ('S0A', 'S0F')

AND A.CUST\_ID = B.PRTN\_ID

AND C.PRTN\_ID = B.PRTN\_ID

AND C.CUST\_CODE = :CUST\_CODE

AND C.CUST\_TYPE = 'C02';

SELECT \*

FROM (SELECT PNM.PRTN\_ID, pnm.prtn\_type

FROM T\_PNM\_PARTNER PNM

CONNECT BY PNM.PRTN\_ID = PRIOR PNM.PARENT\_ID

START WITH PNM.PRTN\_ID IN

(SELECT PRTN\_ID

FROM T\_CUM\_INFO

WHERE CUST\_CODE = :CUST\_CODE)

AND PNM.STAT = 'S0A')

WHERE PRTN\_TYPE = 'PT403';

SELECT A.RULE\_TYPE FROM T\_SCS\_WITHDRAWAL\_RULE A WHERE A.STATUS IN ('S0A','S0F') AND A.CUST\_ID = :CUST\_ID AND A.CUST\_TYPE = 'ORG';

select a.\*,a.rowid from T\_SCS\_WITHDRAWAL\_RULE a;

select a.\*,a.rowid from T\_SCS\_WITHDRAWAL a;

SELECT TO\_CHAR(OPER\_TIME,'YYYYMMDDHH24MISS') AS OPER\_TIME FROM T\_SCS\_WITHDRAWAL WHERE ORDER\_ID = :ORDER\_ID;

SELECT AUDIT\_PERIOD

FROM T\_RSK\_CONFIG

WHERE PRTN\_ID = :PRTN\_ID

AND ACTION\_CODE = :ACTION\_CODE

AND EFF\_DATE < TO\_DATE('%s', 'YYYYMMDDHH24MISS')

AND EXP\_DATE > TO\_DATE('%s', 'YYYYMMDDHH24MISS')

AND DEF\_ID = '16'

AND STAT = 'S0A' ;

select \*

from (SELECT PNM.PRTN\_ID, PNM.BANKROLL\_TYPE, pnm.prtn\_type

FROM T\_PNM\_PARTNER PNM

CONNECT BY PNM.PRTN\_ID = PRIOR PNM.PARENT\_ID

START WITH PNM.PRTN\_ID = :PRTN\_ID

AND PNM.STAT = 'S0A')

where PRTN\_TYPE = 'PT403';

1. **酬金返还程序AutoPayReward**

目录job/New\_AutoPayReward,编译出来的程序名： AutoPayReward，每秒执行（实时程序）。配置文件：config/job.ini,该程序

9.1 程序流程图：



9.2 程序中用到的sql：

获取需要处理的酬金数据：

**实时算实时结，实时算非实时结：**

SELECT \* FROM (SELECT T2.DISC\_INFO\_ID,T2.ORDER\_ID,T2.REWARD\_TYPE,T1.CUST\_ID,T1.TERM\_SEQ,T2.DISC\_ID,T2.DISC\_VALUE,T2.DEAL\_RET FROM T\_SCS\_ORDER T1,T\_SCS\_DISC\_INFO T2 WHERE T1.ORDER\_ID = T2.ORDER\_ID AND T1.STAT = 'S0C' AND T1.PAY\_STAT = 'S0S' AND T1.ACT\_STAT = 'S0S' AND T2.DEAL\_RET = 'S0A' AND T2.REWARD\_TYPE IN ('1002','1003') ORDER BY CREATE\_DATE ASC) WHERE ROWNUM <= 200;

**月结返还酬金：**

SELECT \* FROM (SELECT T2.DISC\_INFO\_ID,T2.ORDER\_ID,T2.REWARD\_TYPE,T1.CUST\_ID,T1.TERM\_SEQ,T2.DISC\_ID,T2.DISC\_VALUE,T2.DEAL\_RET FROM T\_SCS\_ORDER T1, T\_SCS\_DISC\_INFO T2 WHERE T1.ORDER\_ID = T2.ORDER\_ID AND T1.STAT = 'S0C' AND T1.PAY\_STAT = 'S0S' AND T1.ACT\_STAT = 'S0S'

AND T2.DEAL\_RET = 'S0A' AND T2.REWARD\_TYPE = '1001' AND TO\_CHAR(T2.CREATE\_DATE,'MM') = (TO\_CHAR(SYSDATE,'MM')-1) ORDER BY CREATE\_DATE ASC) WHERE ROWNUM <= 200;