Agent Banking and Inhouse Applications Support Manual

Version 1.0

Classified as Internal use only

Contents

Linking, Delinking and Blocking Terminals 3

Failing to Link a Terminal 3

Blocking/Delinking Terminals 3

Delinking from Agentbio 3

Delinking from Agentinfo 3

Unblocking Terminal 3

Unblocking From agentbio 3

Blocking From agentinfo 3

Updating User Branch, Role and User Activation and Deactivation 4

Update User Branch 4

Updating User Role 4

Activating User 5

Deactivating User 5

Checking for transaction status 6

Generating Agents report 8

Generating AGB Exceptions Report Status 10

Supporting mtn cash out 11

Linking, Delinking and Blocking Terminals

Log onto agent banking database server

Failing to Link a Terminal

- select * from agentinfo where agent_id = '23024804';

If agent with terminal id 23024804 exists in **agentinfo** but does not exist in **agentbio,** delete the record from **agentinfo** and request them to link again.

Blocking/Delinking Terminals

Delinking from Agentbio

- select * from agentbio where agent_id = '23024804';

If agent with terminal id 23024804 exists in **agentbio,** run query below.

- update agentinfo set agent_id='23024804 _issueid_492566', account_number ='9030019958956_ issueid_492566', added = false where agent_id = '23024804';

Note: 492566 is the issue tracker id

Delinking from Agentinfo

- Select * from agentinfo where agent_id = '23024804';

If agent with terminal id 23024804 exists in **agentinfo,** run query below.

- update agentbio set agent_id='23024804_issueid_492566', enabled=false, agent_account ='9030019958956 _issueid_492566' where agent_id = '23024804';

Unblocking Terminal

Unblocking From agentbio

- select * from agentbio where agent_id like '%23024804%';

If agent with terminal id 23024804 exists in **agentbio,** run query below.

- update agentbio set agent_id='**23024804'**, enabled=true, agent_account ='The agent's account' where agent_id like '%23024804%';

Blocking From agentinfo

- Select * from agentinfo where agent_id like '%23024804%';

If agent with terminal id 23024804 exists in **agentinfo,** run query below.

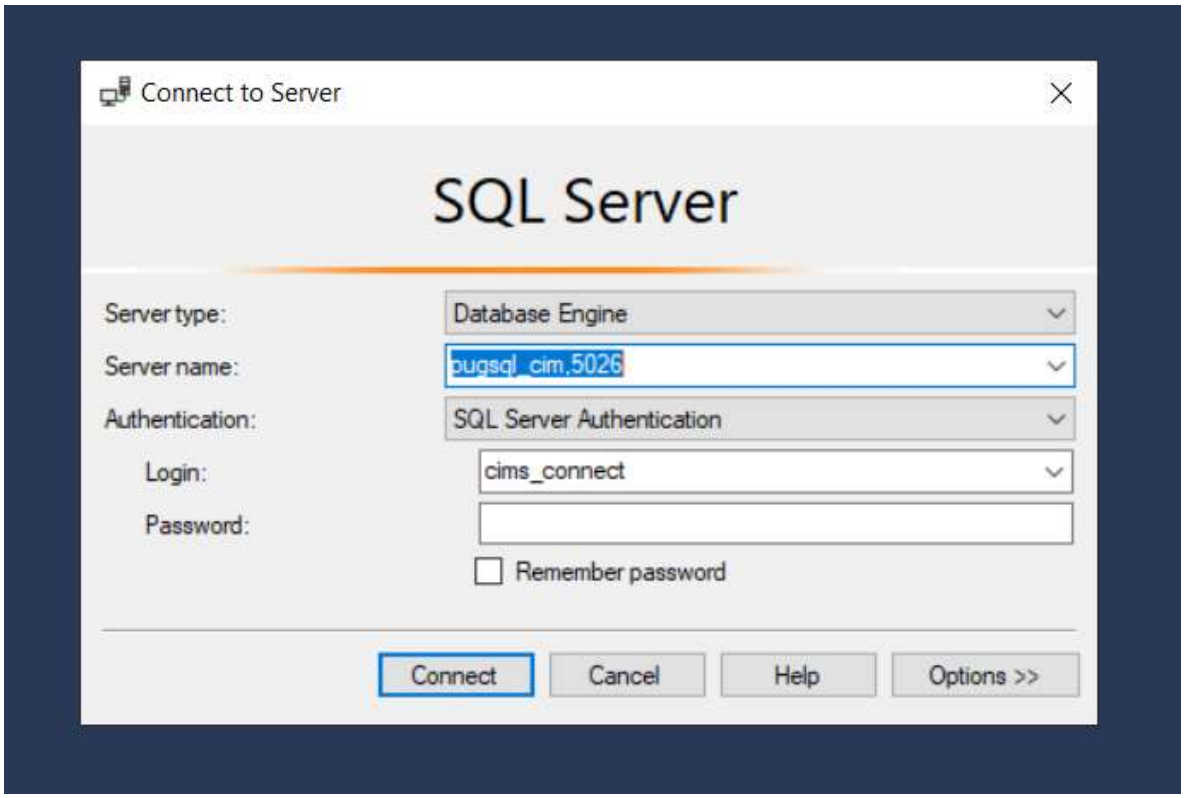
- update agentinfo set agent_id='23024804', account_number ='The agent's account', added = true where agent_id like '%23024804%';

Classified as Internal use only

Updating User Branch, Role and User Activation and Deactivation

Log onto 10.230.205.106 jump box

Open **Microsoft Server SQL Studio** and log onto pugsql_cim database server



Open agent_terminal_linking database and select user_details table

Update User Branch

Select user details using the sap number (userid) as shown in the query below

- select * from user_details where userid = 'a241089';

If user exists, run query below to update the branch

- update user_details set branch = 'gulu' where userid = 'a241089';

Updating User Role

Confirm the various user roles and their ids by right clicking on user_role table and select top 100 from MSSQL to view all the roles or run query below to view all user roles

- select * from user_roles;

Then select user details using the sap number (userid) as shown in the query below

- select * from user_details where userid = 'a241089';

If user with sap number(userid) exists, then run the query below to update user role

- update user_details set role = 1 where userid = 'a241089';

Activating User

Then select user details using the sap number (userid) as shown in the query below

- select * from user_details where userid = 'a241089';

If user with sap number (userid) exists, then run the query below to active

- update user_details set active = 1 where userid = 'a241089';

Deactivating User

Select user details using the sap number (userid) as shown in the query below

- select * from user_details where userid = 'a241089';

If user with sap number (userid) exists, then run the query below to deactivate

- update user_details set active = 0 where userid = 'a241089';

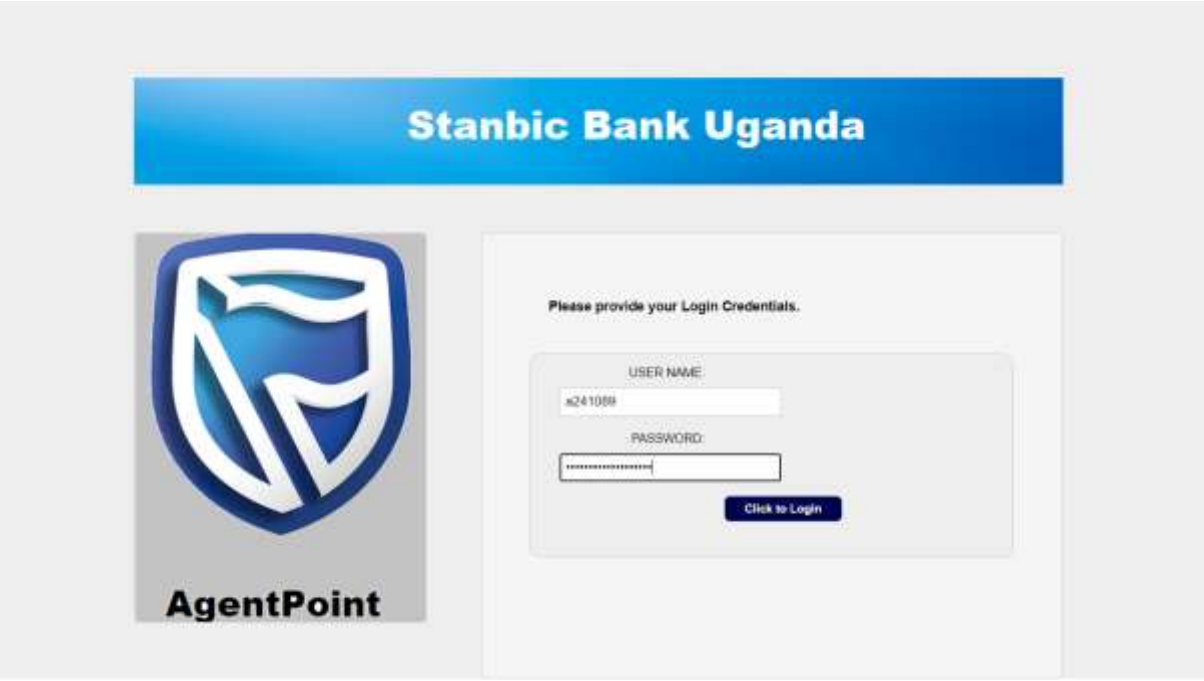
Classified as Internal use only

Checking for transaction status

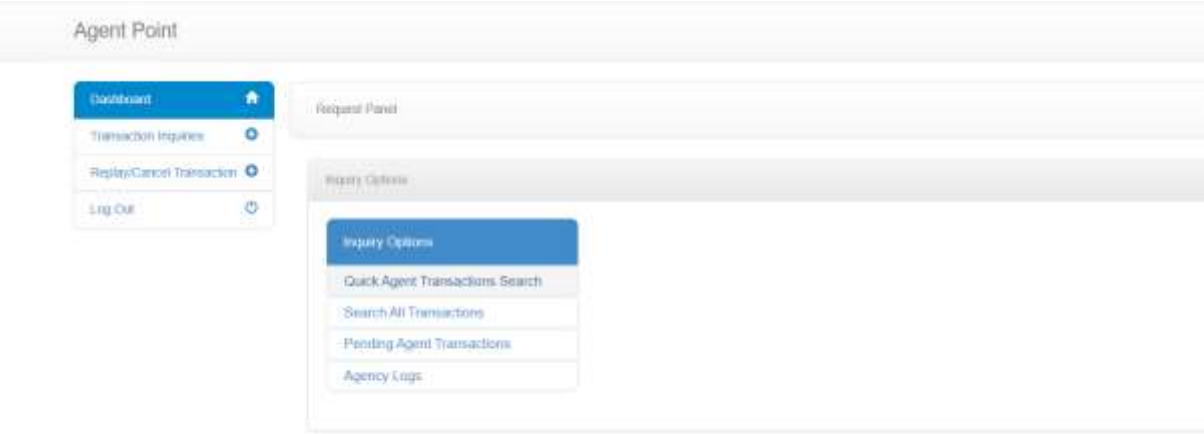
Log onto 10.230.205.106

Open chrome and paste https://pugapp01.ug.sbicdirectory.com/login.aspx

Log into the dashboard with AD credentials



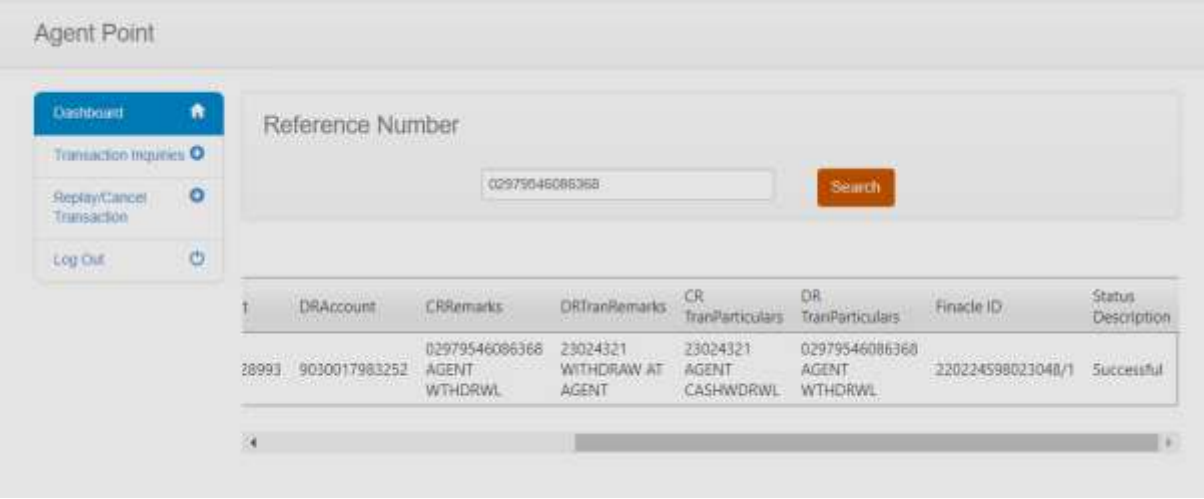
Select 'Transaction inquiries' then select 'Search All Transactions'.



To search for a transaction, add the reference number and click the search button

Details of the transaction will be displayed as below.

If a transaction is successful, status description will show successful



Generating Agents report

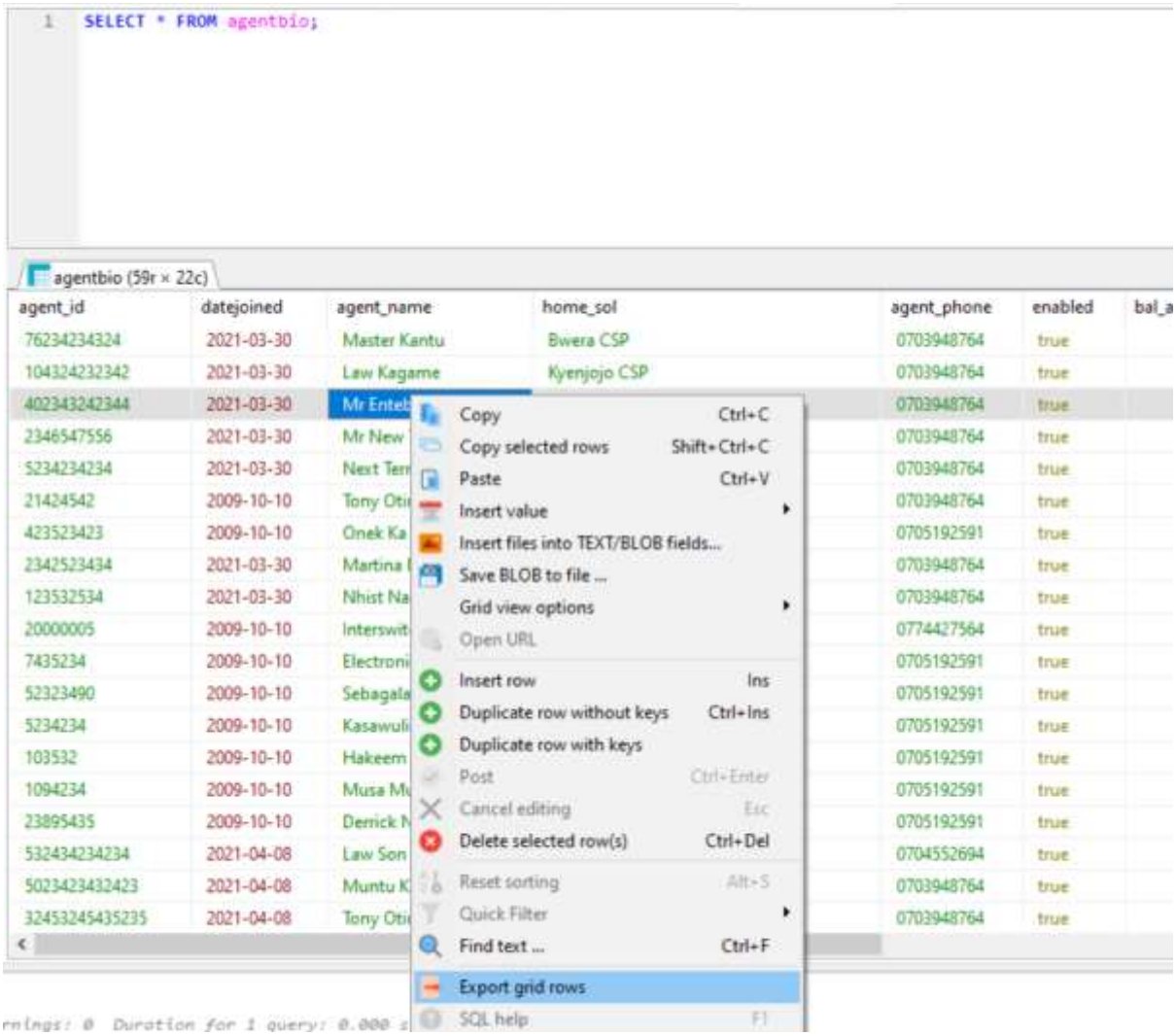
Log onto 10.230.205.106 jump box

Open hiedi sql and log onto the agent banking production database server

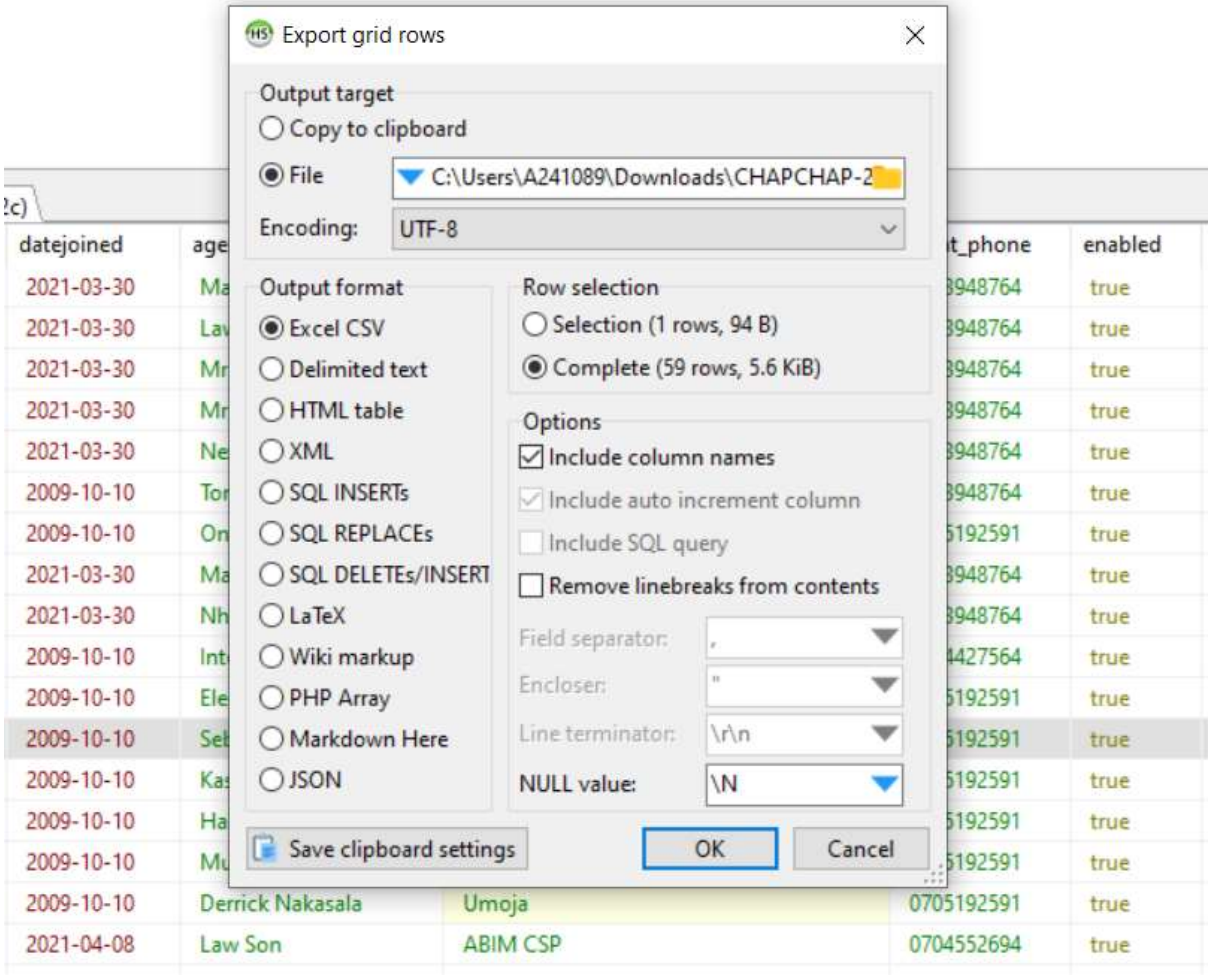
Run query below to select all agents

- select * from agentbio

Right click on the return agents and select export grid rows



Browse where you want to save the file, add a file name and select OK to export agentbio to csv



Classified as Internal use only

Generating AGB Exceptions Report Status

Log onto tranmanager production database server

Select all reference numbers from the shared excel sheet that contains transactions whose exceptions are to be generated

Then run query below to generate an exceptions csv file

- \copy (select instime, request_payload->>'CRAccountNo' CRAccountNo , request_payload->>'DRAccountNo' DRAccountNo, request_payload->>'TranAmount' TranAmount,request_payload->>'RequestID' RequestID, request_payload->>'DRTranParticulars' DRTranParticulars, request_payload->>'CRTranParticulars' CRTranParticulars,response_payload->>'status_desc' status_desc from queued_transactions where requestid in ('010592571528739',
'09801534229702',
'011277034006216',
'07473378865458',
'036242747834439')
) TO '/DBbackup/agbexceptions.csv' with CSV HEADER;

Log onto the **tranmanager** production database server using **winscp** and then download the agbexceptions.csv.

Supporting mtn cash out

Issue

This is an issue which happens whereby an agent funds take long to reflect on their accounts.

Login on to agent banking database server and run the query below to be able to see how many transactions are pending.

The transaction count with approvalresult as **SUCCESSFUL** but with the finstatus as **NULL** refers to transactions which have not been processed yet. But those with approvalresult as **SUCCESSFUL** with finstatus as **P**, it means they are successfully processed.

There are times where the transactions which are waiting to be process are so many and this is how we solve for that issue.

Solution.

Step 1.

Login into agent banking application server.

Step 2.

Run **ps -ef | grep php** to view the running php scripts.

If there a multiple instance of ***php /agentbanking/pugagb/scripts/ resetMTNcredit.php*** running, we need to kill all these processes plus ***php /agentbanking/pugagb/scripts/procMTNLiquidations.php***

as shown in the screen shot below.

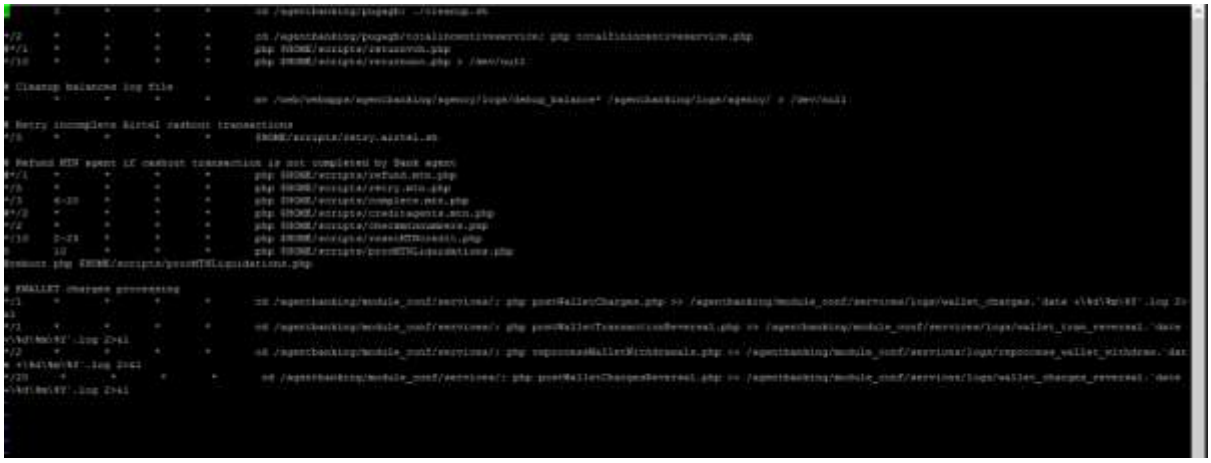
Step3

Run **crontab -e** to edit the existing scheduled cron jobs, particulary

php /agentbanking/pugagb/scripts/ resetMTNcredit.php

php /agentbanking/pugagb/scripts/procMTNLiquidations.php

as shown in the screen short below.



Reschedule the php /agentbanking/pugagb/scripts/procMTNLiquidations.php cron job to re-run again in the next 5 minutes for **procMTNLiquidations.php.**

Note:

1. Please ensure to reschedule the cronjob back to the default value which is supposed to be **2 am** when it is supposed to restart as soon as the backlog has been cleared.
2. **procMTNLiquidations.php** process must be killed first before it can be rescheduled to run in the next 5 minutes. Failure to do this will result in multiple instances of the same service running at the same time which can potentially result into duplicates.
3. **procMTNLiquidations.php** should never be executed manually when the original process is already running as this can also result into the issue highlighted in 2: above.
4. Killing of these processes can be done a couple of times until all the transactions have been successfully processed as you monitor from the database side.

Classified as Internal use only