
CANCEL INVOICE

General Notes :

(1) Deposit Cancellation not allowed (Just release/rollback allowed) and deposit status will change from unpaid to paid just once and no change after, Deposit will be handled manually.

(2) @ invoice cancellation, user will be prompt of all payments related to invoice (including estimated-cancellation-type per Payment) and this panel will be refreshed every minutes if no action taken.

(3) while cancelling invoice first active payments will be cancelled then [sub]deposits will be release and then invoice will be reversed and tagged as cancelled
if crashing happens at payments, successful cancelled-payments will be effective and does not go further

Logic Summary:

- (1) CALL **[invoicePaymentsCancellationPossibility]** TO CHECK ACTION IS POSSIBLE [server-side]
- (2) Cancels All Payments done for this Invoice [INTERNAL/CC/DB] [server-side external-system]
- (3) Call **[cancel_invoice]** to Roll-backs All deposits attached to Invoice and rollback invoice transactions

Logic :

- 1-DESCRIPTIONS
- 2-IF INVOICE IS DRAFT OR CANCELLED THEN EXIT
- 3-FETCH INVOICE INFO
- 4-RELATED ACTIVE[SUB]DEPOSITS ROLLBACK ACTION, ATTACHED TO THIS INVOICE IF ANY EXISTS **[rollbackDeposit]**
- 5-RELATED INVOICE PAYMENTS CANCELLATION --COMMENTED
- 6-GET ACCOUNTS
- 7-RECORD ACTION
- 8-INVOICE TRANSACTIONS ROLLBACK
- 9-UPDATE INVOICE TO STATUS=CANCELLED

[releaseDeposit](master/slave/currency/invoice/_ReqAmount)

general Notes:

- (1) release deposit from dep account to wallet account fully or partially based on master/slave/currency
- (2) only deposit made between master/slave would be releasable to invoice between master/slave no matter of ChargeType
- (3) if requested amount is null it means there is no limit for amount released

Logic:

- 1-GET DESCRIPTIONS
- 2-GET ACCOUNTS
- 3-ALL PAID DEPOSIT MADE FROM INVOICE RECEIVER TO INVOICE ISSUER
loop
- 3-1-CHECKUP IF INVOICE IS FULLY PAID SO FAR THEN EXIT LOOP
- 3-2-GET REAL AND NAMED DEPOSIT
- 3-3-IF THIS DEPOSIT DOES NOT HAVE ROOM THEN CONTINUE (FAKE DEPOSIT SPENT ALL

THEN USE OTHER LEFT DEPOSITS)
 3-4-GET MAXIMUM ALLOWABLE RELEASABLE AMOUNT
 BETWEEN REQUESTED_AMOUNT & OWING & THIS DEPOSIT FRAME
 3-5-GET ADJUSTABLE AMOUNT NEED TO BE TAKEN OFF FROM
 BOTH CURRENT REAL AND NAMED DEPOSIT, NOT TO EXCEED CURRENT NAMED DEPOSIT LIMIT
 3-6-RECORD ACTION
 3-7-RECORD DEPOSIT USAGE FOR BOTH NAMED AND REAL DEPOSIT PORTION
 3-8-TRANSACTIONS : OPEN THE DEPOSIT IF UNTOUCHED OR ROLLBACKED
 3-9-TRANSACTIONS
 3-10-UPDATE INVOICE AMOUNTS AND IF IS FULLY_PAID THEN CHANGE STATUS TO PAID
 3-11-CHECKING IF REQUESTED_AMOUNT DID NOT USE THE WHOLE CURRENT DEPOSIT FRAME
 THEN THIS DEPOSIT FRAME WOULD BE BE LAST FRAME WE CHECK
 end loop;

[rollbackSubDeposit]

General notes:

Gets invoice_subdeposit record and detaches from invoice by disabling that

1-GET DESCRIPTIONS
 2-GET INVOICE_ID
 3-RECORD ACTION
 4-FOR ALL TRANSACTIONS BELONG TO INVOICE_SUBDEPOSIT
 loop
 4-1-INSERT REVERSE TRANSACTIONS
 4-2-SAVE A COPY OF ROLLED BACK TRANSACTION IN SUBDEPOSIT_TRANSACTION
 end loop;
 5-DETACH SUBDEPOSIT FROM INVOICE BY DISABLING THAT
 6-UPDATE INVOICE AMOUNTS BY THE AMOUNT RELEASED FROM INVOICE

uses deposit_required and invoice_subdeposit and subdeposit_txn tables

[invoicePaymentsCancellationPossiblity](invoice_id)

General Notes :

(1)IF SUM(Invoice [None Cancelled] Payments + SUM(cancellation_fees_per_payment)) <= Master_entity_wallet then true

(2)because of time-based cancellation fees, this method would be called 2-times per invoice cancellation request, first at prompting second time when its going to be effective

[get_deposit_balance](master,slave,currency) 0208

general Notes:

(1)Gets total Paid Deposit Made - Used Active Deposits Amount (between Master/Slave)

[get_deposit_releasable](deposit_id,real::boolean) 0208

General Notes:

logic :

1-GET DEPOSIT REAL/UNREAL DEPOSITS
2-GET ALL REAL DEPOSIT PORTION AMOUNTS USED IN THIS DEPOSIT
3-FOR BOTH RELEASBLE REAL/UNREAL DEPOSIT , CHECK IF AMOUNT >= AMOUNT_USED THEN
RETURN DIFF ELSE RETURN 0

[release_deposit_invoice](invoice_id,amount) 0207

--GET INVOICE ENTITIES
--GET REMAINING RELEASABLE DEPOSITs LEFT [get_deposit_balance]
--RELEASE DEPOSIT IF REQUESTED AMOUNT IS LESS THAN REAEASEABLE_DEPOSITS CALL TO [releaseDeposit]

Sample

Deposit = 100
Real Deposit = 97

Request patterns minimum(Owe,reqAmt,remainingPortionOfDeposit)

(RPM#1)

50 >> [get_deposit_releasable(f/t)] (*)50 (**)50
50 >> [get_deposit_releasable(f/t)] (*)50 (**)47
01 >> [get_deposit_releasable(f/t)] (*)00 X

(RPM#2)

100 >> [get_deposit_releasable(f/t)] (*)100 (**)97
01 >> [get_deposit_releasable(f/t)] (*)00 X

(RPM#3)

98 >> [get_deposit_releasable(f/t)] (*)98 (**)97
02 >> [get_deposit_releasable(f/t)] (*)02 (**)00
01 >> [get_deposit_releasable(f/t)] (*)00 X

(RPM#4)

99 >> [get_deposit_releasable(f/t)] (*)99 (**)97
01 >> [get_deposit_releasable(f/t)] (*)01 (**)00
01 >> [get_deposit_releasable(f/t)] (*)01 (**)00
01 >> [get_deposit_releasable(f/t)] (*)00 X

(RPM#5)

50 >> [get_deposit_releasable(f/t)] (*)50 (**)50
48 >> [get_deposit_releasable(f/t)] (*)48 (**)47
02 >> [get_deposit_releasable(f/t)] (*)02 (**)00
01 >> [get_deposit_releasable(f/t)] (*)00 X