

Undoing a Bank Reconciliation

09/13/2024 6:17 pm EDT

Issue:

A customer is asking to have a Bank Reconciliation deleted so it can be redone.

How is this done in SedonaOffice?

Resolution:

UNDOING OR REVERSING A BANK RECONCILIATION in nine easy steps.

Always test this in a sandbox first so the customer can verify the data after the change.

Undoing a Bank Rec. will undo the bank reconciliation back to the date of the of the problem bank rec.

Below are the steps in more detail.

1) Have the user refresh their Sandbox to match production data; if SedonaCloud, our IT can do this.

2) Check the max timestamp to be sure you're working with current data */

```
SELECT MAX(Date_Time_Stamp) FROM GL_Register
```

3) Know the Bank Account Code.

```
SELECT * FROM GL_Account WHERE Account_Code = '--AcctId
```

4) Backup tables.

```
SELECT * INTO GLRegBKUPTktABCDEF FROM GL_Register
```

```
SELECT * INTO ARDepBKUPTktABCDEF FROM AR_Deposit
```

```
SELECT * INTO GLAcctReconBKUPTktABCDEF FROM GL_Account_Reconcile
```

```
SELECT * INTO AccRecLogBKUPTktABCDEF FROM Account_Reconciliation_Log
```

5) Identify the starting point based on your Account Reconcile table; you will be updating GL and Deposit data for all from that point in time (and greater):

****NOTE:** generally, you can look at the Starting_Reconcile_Date and judge... but some customers do not do monthly reconciliations, they do them daily which makes this a bit more difficult.. In this case you have to make a judgment call based on the data you're seeing. They might also do duplicates and not always 1 per month which throws off their reconciliation totals.

```
SELECT * FROM GL_Account_Reconcile WHERE Account_ID = x ORDER BY Account_Reconcile_Id DESC
```

```
SELECT * FROM Account_Reconciliation_Log WHERE Account_Id = x OR Bank_Account_Id = x ORDER BY EntryDate DESC
```

6) Set statuses back to open, based on EntryDate (identified from step 5).

```
DECLARE @EntryDate datetime
```

```
DECLARE @AcctId int
```

```
SET @EntryDate = 'YYYY-MM-DD'
```

```
SET @AcctId = x
```

```
UPDATE d
```

```
SET Reconciled = ''
```

```
FROM Account_Reconciliation_Log l
```

```
INNER JOIN AR_Deposit d ON d.Register_ID = I.Deposit_Register_Id
```

```
WHERE I.EntryDate >= @EntryDate AND I.Bank_Account_Id = @AcctId
```

```
UPDATE r
```

```
SET Status = 'O'
```

```
FROM Account_Reconciliation_Log I
```

```
INNER JOIN GL_Register r ON r.Register_ID = I.Register_Id
```

```
WHERE I.EntryDate >= @EntryDate AND I.Account_Id = @AcctId
```

7) Delete the account reconciliation records from the table.

```
DELETE GL_Account_Reconcile WHERE Account_Reconcile_Id IN (xyz)
```

8) Be sure nothing in R status; R means it's been checked and not-yet-reconciled.

```
UPDATE GL_Register SET Status = 'O' WHERE Register_Id IN (SELECT Register_Id FROM GL_Register WHERE Status NOT IN ('O','C') AND Account_Id = x) -
```

```
UPDATE AR_Deposit SET Reconciled = '' WHERE Bank_Account_Id = x AND Reconciled NOT IN ('',NULL,'C','O')
```

9) Get from customer what the Last Ending should be and update it accordingly.

```
UPDATE GL_Account SET Last_Ending_Balance = xxx.xx WHERE Account_ID = XX
```