# ez Bank Reconciliation

# USER MANUAL

& TECHNICAL MANUAL

Getting Started	5
Before you begin	5
Brief overview	5
Get familiar with the interface	5
Navigation	5
Dashboard	6
Reconciliation Area	8
Match Transactions	8
Reporting	14
Matched Transaction Details	
To un-match transactions	
Search For Transactions	
Staging Area	
Closing Balances	
Import Transactions	21
Bank and Ledger Staging Screens	23
Maintenance Area	25
Accounting Period	25
SS Clearance File	
Adjustments	
View Process History	
Administrator Area	
Bank and Ledger Groups	
Custom Processes	
Custom Reports	
Users and Access	
Bank Accounts	
Advanced Configuration	
Database Overview	
Account Related Tables	
Stored Procedures	
Application Settings	

Technologies & Resources	40
.Net Blazor	
MudBlazor	
Closedxml	40
Closedxml.Reports	
Dapper	40
Signal R	
Excel Data Reader	
Microsoft SQL Server	
Installation	
Requirements	
Sample Installation	
Database Schema	
APPENDIX 1 – Scripts for Main Tables	
account_bank_file_mapping	
account_ledger_file_mapping	
processes	
roles	
user_accounts	
users	
APPENDIX 2 – Scripts for Account Tables	
accounting_periods	
bank_transaction_groups	
bank_transactions	53
bank_transactions_staging	53
closing_balances	54
ledger_transaction_groups	
ledger_transactions	
ledger_transactions_staging	
process_logs	
user_bank_columns	
user_ledger_columns	
custom_processes	

reports_history	60
APPENDIX 3 – Scripts for Views	61
ledger_closing_balance_vw	61
ledger_trans_sum_vw	61
bank_trans_sum_vw	61
APPENDIX 4 – Scripts for Stored Procedures	61
bank_staging_process	
import_ledger_transactions	
ledger_staging_process	
generate_ss_fILe	
matching_process	

# **Getting Started**

#### Before you begin

Before you begin using the software, please ensure that the software is installed and configured properly. (See **Installation Section at End of this Guide**) Once, installed, use your browser and navigate to the url.

#### Brief overview

The EzBankRecon software provides a simple way to manage the reconciliation of accounts. Primarily it was designed to work with the SmartStream system. This manual will assist you with getting familiar with the functionality of the program.

# Get familiar with the interface

#### Navigation

The main navigation of the program is located to the left of the page.



Note. The menu items that appear is based on the user security.

Menu	Role
Reconciliation	Regular User
Staging	Super User
Maintenance	Super User
Administration	Administrator

Next area of interest is the top bar.

The top bar shows the active account, the user who logged in, and also a shortcut to log out. The hamburger menu closes or opens the main navigation of the program

ezBankRecon		≡ National Bank – 86	55588441	Kelvin Woodman 🕐 [+
Home		Switch Bank Account		
Reconciliation	*			
*** Staging				
🗣 Maintainance	*			
G Administration		National Bank	Scotia Bank	
[→ Log Out		SELECT	DASHB	OARD
		Previous P	eriod: Feb 2023_og	Current Period: Jan 2023 _og
			Dashboard	

To Access the Dashboard click on the Home menu. The dashboard shows information on the last two periods of reconciliation. It also serves as the place where you can switch accounts if you have access to multiple bank accounts.

ezBankRecon		■ National Bank – 865	55588441	Kelvin Woodman 🔗 [+
f Home		Switch Bank Account		
Reconciliation	*			
;∷* Staging				
🗣 Maintainance	*			
G Administration	.*	National Bank	Scotia Bank	
[→ Log Out		SELECT	SELECT	
	-		DASHB	OARD
		Previous P	eriod: Feb 2023_og	Current Period: Jan 2023 _og

If you only have access to one account, the "Switch Bank Account" option will not show on the dashboard. Also, if you have multiple accounts, the **menu items** will not show unless you select an account to work with.

The dashboard also provides information on whether all the imported transactions match with the balance. A "Discrepancy" message is issued if the balances do not match. Discrepancies may arise due to not all transactions being uploaded, or wrong closing balance amount were entered.

	DASHI	BOARD	
Previous Perio	d: Feb 2023_og	Current Period	l: Jan 2023 _og
Bank Closing Balance \$420.00 Sum of Uploaded Transactions \$7,269.82 Discrepency Found!	Ledger Closing Balance \$45.00 Sum of Uploaded Transactions (\$241,653.06) Discrepency Found!	Bank Closing Balance \$123.00 Sum of Uploaded Transactions \$7,269.82 Discrepency Found!	Ledger Closing Balance \$123.00 Sum of Uploaded Transactions (\$241,653.06) Discrepency Found!
Unmatched Bank Summary Credit Transactions \$258.36 Debit Transactions	Unmatched Ledger Summary Debit Transactions (\$356,089.93)	Unmatched Bank Summary Credit Transactions \$7,269.82	Unmatched Ledger Summary Debit Transactions (\$241,653.06)

# **Reconciliation Area**

Let's look at the Reconciliation Menu.



There are 4 sub-menu items.

- Match Transactions
- Reporting
- Matched Trans. Details
- Search For Transactions

#### **Match Transactions**

This is where majority of the work will be done. Let's familiarize ourselves with the interface.

$\equiv$ National Ban	k — 8655588441				Kelvin Woodman 🎯 [→
Transaction Mate	ching			E) 🎽	± ≠ 10
BANK TRANSACTIONS	Filter By Groups	• 🗉 🛓 🖬	LEDGER TRANSACTIONS	Filter By Groups	• 🗉 🔌 💷
Actions	Grouping Y	Amount Y	Actions	Amount Y	Journal Id マ
	Deposits	\$7,269.82		(\$65,000.00)	ADJF5001067
	Transfers	\$7,269.82	D 🖽 📀 📖	(\$148,678.06)	ADJF5001062
	Transfers	\$258.36		\$100.00	EZADJF500389
	Payments	\$7,269.82	D 🖽 📀 🗐	(\$28,075.00)	ADJF5001643
D 🖽 📀 🗐	Transfers	\$258.36		(\$88,058.52)	ADJF5001063
D 🖽 📀 💷	Transfers	\$7,269.82	D 🖽 📀 📖	(\$100,000.00)	ADJF5001066
	Transfers	\$7,269.82		\$43,736.36	ADJF5001365
D 🖽 🕜 🗐	Payments	\$7,269.82	D 🖽 📀 🗐	(\$28,675.00)	ADJF5001644
	Transfers	\$7,269.82		(\$75,348.51)	ADJPAYY100168
BANK 0 E Amou	nt \$0.00	> MATCH TRA	NSACTIONS «	LEDGER 0 An	nount \$0.00

The top bar with the title "Transaction Matching" contains the following functionality:

Icon	Function					
	Adjust the grid size to fit the screen. (Only necessary if the system failed to adjust it automatically)					
<b>*</b> *	This runs the auto-mate	ching process				
<u>+</u>	Pressing this button do sheet	wnloads the unmatche	d bank and le	dger transacti	ions to an excel	
P*	This runs any customs p	processes that the adm	inistrator deve	eloped		
			lavoute for th	e bank and le	daer arids.	
11	This toggles between I Horizontal View (Side	by Side)				
11	This toggles between I Horizontal View (Side BANK TRANSACTIONS Filter By Gro	by Side)	LEDGER TRANSACTIO	Filter By Groups	- <u> </u>	
11	This toggles between I Horizontal View (Side BANK TRANSACTIONS Filter By Gro Actions Amount 4 1	ups - Qrouping V	LEDGER TRANSACTION	ONS Filter By Groups Amount Y	IIII 🛓 🗐 🗸 IIII	
11	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS         Filter By Gro         Actions       Amount \$1.1         Image: Comparison of the structure of the structur	ups  Grouping  Chapter  Chapte	LEDGER TRANSACTIV Actions	ONS     Filter By Groups       Amount     V       (\$65,000.00)	→ □ <u>&gt;</u> Ⅲ Journal Id ADJF5001067	
11	This toggles between I Horizontal View (Side BANK TRANSACTIONS Filter By Gro Actions Amount 4 1 *	Ups Crouping V Grouping V Deposits Transfers	LEDGER TRANSACTION	ONS         Filter By Groups           Amount         T           (\$65,000.00)         (\$148,678.06)	→ □ <u>&gt;</u> □ <u>&gt;</u> □ Journal Id <u>\</u> ADJF5001067 ADJF5001062	
11	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS         Filter By Gro         Actions       Amount 4 1 °         C       S7,269.82         C       S7,269.82         C       S7,269.82	porizontal and vertical by Side) ups - Q & M Grouping Q Deposits Transfers Payments	LEDGER TRANSACTION	ONS         Filter By Groups           Amount         T           (\$65,000.00)         (\$148,678.06)           \$100.00         \$100.00	Journal Id         V           ADJF5001067         EZADJF500389	
11	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS         Filter By Gro         Actions       Amount 4 1 1         Image: Comparison of the system         Image: Compariso	orrizontal and vertical by Side) ups → Q & M Grouping ♥ Deposits Transfers Payments Transfers	LEDGER TRANSACTIV Actions	ONS         Filter By Groups           Amount         Y           (\$65,000.00)         (\$148,678.06)           \$100.00         (\$28,075.00)		
11	Actions       Amount 4 1 %         Image: Control of the con	Ups vertical by Side) ups vertical Grouping vertical vertical vertical ups vertical verti	LEDGER TRANSACTIV Actions	ONS         Filter By Groups           Amount         ♥           (\$65,000.00)         (\$148,678.06)           \$100.00         (\$28,075.00)           (\$88,058.52)         (\$88,058.52)	Journal Id EZADJF5001062 ADJF5001063 ADJF5001063	
11	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS         Filter By Gro         Actions       Amount ↓ 1 °         □<	by Side) ups  Grouping Groupin	LEDGER TRANSACTIV Actions C EL O C C E C O C	CNS         Filter By Groups           Amount         ♥           (\$65,000.00)         (\$148,678.06)           \$100.00         (\$28,075.00)           (\$88,058.52)         (\$100,000.00)	Journal Id         V           Journal Id         V           ADJF5001067         ADJF5001062           EZADJF500389         ADJF5001643           ADJF5001063         ADJF5001063           ADJF5001063         ADJF5001063	
	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS       Filter By Gro         Actions       Amount 4 1 °         C       C       S7,269.82	ups     Image: Complexity of the complex	LEDGER TRANSACTIV Actions C EL O III C EL O IIII C EL O IIIII C EL O IIIIII C EL O IIIIII C EL O IIIIII C EL O IIIIII C EL O IIIIIII C EL O IIIIII C EL O IIIIII C EL O IIIIII C EL O IIIIIII C EL O IIIIIIII C EL O IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Filter By Groups           Amount         V           (\$65,000.00)         (\$148,678.06)           (\$148,678.06)         \$100.00           (\$28,075.00)         (\$28,075.00)           (\$88,058.52)         (\$100,000.00)           \$43,736.36         \$43,736.36	Journal Id Journal Id ADJF5001067 ADJF5001062 EZADJF500389 ADJF5001643 ADJF5001063 ADJF5001066 ADJF5001066	
	This toggles between I         Horizontal View (Side         BANK TRANSACTIONS       Filter By Gro         Actions       Amount 4 1 °         Q<       \$7,269.82         E       \$7,269.82	ups     Image: Comparison of the compari	LEDGER TRANSACTIV Actions C EL 0 C III C EL 0 III III	CNS         Filter By Groups           Amount         Y           (\$65,000.00)         (\$148,678.06)           \$100.00         (\$28,075.00)           (\$88,058.52)         (\$100,000.00)           \$43,736.36         (\$28,675.00)	Journal Id         V           ADJF5001067            ADJF5001062            EZADJF500389            ADJF5001063            ADJF5001063            ADJF5001063            ADJF5001063            ADJF5001063            ADJF50010643            ADJF5001065            ADJF5001066	

BANK TRANSACTION	S	Fi	ter By Groups		
Actions	Amount $\downarrow$ 1 $\heartsuit$	Grouping Y	Posting Date 7	Description 1	₽ De
	\$7,269.82	Deposits	2023-01-02	XML02301020000022 102808	02 P.A.Y.E Tax Re
	\$7,269.82	Transfers	2023-01-02	02301020000022 10280802 P.	A.Y.E Tax Re
	\$7,269.82	Payments	2023-01-02	02301020000022 10280802 P.	A.Y.E Tax Re
LEDGER TRANSACTIO	\$7,269.82	Payments	2023-01-02	02301020000022 10280802 P.	A.Y.E Tax Re
LEDGER TRANSACTION	\$7,269.82	Payments F Journal Id V	2023-01-02	02301020000022 10280802 P	A.Y.E Tax Re
LEDGER TRANSACTION Actions	\$7,269.82	Journal Id V ADJF5001067	2023-01-02	02301020000022 10280802 P	A.Y.E Tax Re
LEDGER TRANSACTION Actions	\$7,269.82 ■ Amount ♀ (\$65,000.00) (\$148,678.06)	Payments           Journal Id         Y           ADJF5001067         ADJF5001062	2023-01-02 ilter By Groups Effective Date ♥ 2022-07-31 2022-07-31	02301020000022 10280802 P.           Construction           Construction	A YE Tax Re

Now, let's move to the Bank and Ledger data grids.

BANK TRANSACTIONS	Filter By Groups	• 🗉 💁 💷	LEDGER TRANSACTIONS	Filter By Groups	• 🗉 🛓 🖬
Actions	Grouping Y	Amount 🖓	Actions	Amount Y	Journal Id 🛛 🍸
	Deposits	\$7,269.82		(\$65,000.00)	ADJF5001067
	Transfers	\$7,269.82		(\$148,678.06)	ADJF5001062
	Transfers	\$258.36		\$100.00	EZADJF500389
000	Payments	\$7,269.82	D 🖽 📀 📖	(\$28,075.00)	ADJF5001643
	Transfers	\$258.36		(\$88,058.52)	ADJF5001063
0 🖽 📀 📖	Transfers	\$7,269.82		(\$100,000.00)	ADJF5001066
	Transfers	\$7,269.82		\$43,736.36	ADJF5001365
	Payments	\$7,269.82		(\$28,675.00)	ADJF5001644
- • •	Transfers	\$7,269.82	D 🖽 🔗 📖	(\$75,348.51)	ADJPAYY100168

The top of the grids provide a quick filter, whereby the list will be filtered by a group selected. The icons provide the following functionality.

lcon	Functionality	
Ę	Insert user comment on all selected transactions are selected.	rows. This button will remain disabled until one or more
	Comments	×
	Comments	
		CANCEL SAVE
<u>è.</u>	This changes the row color of all se or more transactions are selected.	lected rows. This button will remain disabled until one

Row Color	×			
	CANCEL SAVE			
 Change the colum grids) The column selecti	ns being displayed on screens is as fol	d. (Note. Your prefe lows: Display the following Ledg	rence is saved and will a er Columns ×	iffect all
		Grouping	Description	
Display the following Bank C	olumns	Trans. Amount     Journal Id     Ledger id	Date Posted      Payment Method      Payment Ref. No.	
Grouping	Contraction 1	Effective Date	Dayment Ref. Date	
Branch	Description 2	Drnl Sequence no.	Dendor Id	
Dustomer	Cescription 3	Ministry	Dendor Location	
Decount No.	Description 4	De Program	Dendor Name	
amount	Description 5	Sub Program	Dank id	
Posting Date	Description 6	Account	Bank Acct. No.	
Value Date	Desting No.	D Project	Dayable Entity	
Clearing Date	Accounting Period	D SOF	Payment Request	
Transaction No.	Comments	Sector	Payment Amount	
	CANCEL	Posting Year     Posting Period	Payment type	

#### Next, let's look at the grid header.

BANK TRANSACTIONS	Filter By Groups	• 🗉 <u>è.</u> 💷	LEDGER TRANSACTION	Filter By Groups	• 🗉 🛓 🖩
Actions	Grouping Y	Amount 🛛 🍸	Actions	Amount Y	Journal Id 🛛 🏹
	Deposits	\$7,269.82		(\$65,000.00)	ADJF5001067
	Transfers	\$7,269.82	D 🖽 🔗 📖	(\$148,678.06)	ADJF5001062
	Transfers	\$258.36	D 🖽 🔗 📖	\$100.00	EZADJF500389
	Payments	\$7,269.82		(\$28,075.00)	ADJF5001643
🗆 🖽 📀 📖	Transfers	\$258.36	D 🖽 🕜 🗏	(\$88,058.52)	ADJF5001063
	Transfers	\$7,269.82	D 🖽 🔗 📖	(\$100,000.00)	ADJF5001066
	Transfers	\$7,269.82	D 🖽 🕜 🗐	\$43,736.36	ADJF5001365
	Payments	\$7,269.82		(\$28,675.00)	ADJF5001644
	Transfers	\$7,269.82		(\$75,348.51)	ADJPAYY100168

The header allows you to sort, filter and move columns around as you see fit. To sort, you can just click on the column name. There is robust filtering option as seen in the following image.

BAN	K TRANSACTIONS	Filter By	Groups	• 🛛	<u> </u>	LEDGER TRA
Actio	ons	Amount	Y	Grouping	Ŷ	Actions
	Column		Operator	,	/alue	5
×	Amount			*	Filter value	<u>^</u>

Next let's look at the row functionality.

BANK TRANSACTIONS	Filter By Groups	• 🗉 🖄 💷	LEDGER TRANSACTIONS	Filter By Groups	• 🗉 <u>è.</u> 💷
Actions	Grouping 🖓	Amount Y	Actions	Amount Y	Journal Id 🛛 🍸
	Deposits	\$7,269.82		(\$65,000.00)	ADJF5001067
	Transfers	\$7,269.82		(\$148,678.06)	ADJF5001062
	Transfers	\$258.36		\$100.00	EZADJF500389
	Payments	\$7,269.82		(\$28,075.00)	ADJF5001643
	Transfers	\$258.36		(\$88,058.52)	ADJF5001063
	Transfers	\$7,269.82		(\$100,000.00)	ADJF5001066
	Transfers	\$7,269.82		\$43,736.36	ADJF5001365
	Payments	\$7,269.82	D 🖽 🍞 📖	(\$28,675.00)	ADJF5001644
	Transfers	\$7,269.82		(\$75,348.51)	ADJPAYY100168

The Actions in the grid provide the following functionality.

lcon	Functionality				
ſ	If there are mu icon will show. icon it will show off. Also, an ac	ltiple users worki You will not be a v you who has loc dministrator can u	ng on an accour ble to select the ked the record. mlock items	nt, the another u It record. If you NOTE: locked r	ser selects a record, the loc hover your mouse over the ecords are cleared on log
	BANK TRANSACTIONS	Filter By Groups	- E <u>&amp;</u> III	LEDGER TRANSACTIONS	Filter By G
	Actions	Grouping 🖓	Amount 🛛 🍸	Actions	Amount
		Deposits	\$7,269.82	D 🕀 😗 🗐	(\$65,000.00)
		Transfers	\$7,269.82		(\$148,678.04
	🗆 🖽 📀 📖	Transfers	\$258.36	0 🖽 🕜 📖	\$100.00
	<b>R</b> E () E	Payments	\$7,269.82		(\$28,075.00)
	<b>R</b> = 0 =	Transfers	\$258.36	0 🖽 📀 📖	(\$88,058.52)
	<b>R</b> E O E	Transfers	\$7,269.82		(\$100,000.00
	<b>E</b> 🕐 🗐	Transfers	\$7,269.82		\$43,736.36
		10000000000000000000000000000000000000	20000000000		(400 675 00)

	Comments ×
	Comments
	CANCEL SAVE
0	Changes the row grid color based on a preset list of colors
	Hovering over this provides a quick view of the user comment on the transaction. (Note. This will be disabled if no comments are present)

Now let's look at the last area, the summary and transaction matching.

■ National Banl	< — 8655588441				Kelvin Woodman ⊘ [
Transaction Mate	cning			L. 🐣	🛎 🖌 [1]
BANK TRANSACTIONS	Filter By Groups	• 🗏 🛓 🖬	LEDGER TRANSACTIONS	Filter By Groups	• 🗉 🛓 🖬
Actions	Grouping Y	Amount 🛛	Actions	Amount 🛛	Journal Id 🛛 🏹
0 🖽 🤭 🗐	Deposits	\$7,269.82		(\$65,000.00)	ADJF5001067
	Transfers	\$7,269.82	D 🖽 📀 💷	(\$148,678.06)	ADJF5001062
	Transfers	\$258.36		\$100.00	EZADJF500389
D 🖽 📀 📖	Payments	\$7,269.82	D 🖽 📀 📖	(\$28,075.00)	ADJF5001643
D 🖽 📀 🗐	Transfers	\$258.36		(\$88,058.52)	ADJF5001063
- • •	Transfers	\$7,269.82		(\$100,000.00)	ADJF5001066
) 🖽 📀 🔳	Transfers	\$7,269.82	D 🖽 📀 📖	\$43,736.36	ADJF5001365
I 🖽 📀 💷	Payments	\$7,269.82		(\$28,675.00)	ADJF5001644
D 🖽 📀 🗐	Transfers	\$7,269.82	🗆 🖽 🤣 🗐	(\$75,348.51)	ADJPAYY100168
ANK 0 🗖 Amou	Transfers	\$7,269.82		(\$75,348.51)	ADJPAYY100168 mount \$0.00

The number in the box shows how many transactions are selected.

The - button next to the "Amount" clears all selected transactions.

The Amount shows you the sum of the selected transactions.

The "Match Transactions" button, allows you to match the transactions. The button will only be enabled when the sum of the selected transactions are zero. Once transactions are "matched", they are removed from the list.

#### Reporting

Select Reporting under the menu. This screen handles all the reporting needs of the program.

Under the reporting section, there are three options. Built-In Reports, Historical Reports and User Reports.

The Built-in Reports allows you to run the Reconciliation Statement, along with items that have not been reconciled. This is the only report that can be saved.

Reporting		
BUILT-IN REPORTS HISTORICAL REPORTS	Report Reconciliation Statement	•
USER REPORTS	CURRENT PERIOD: Jan 2023 _og	

All Built-in reports are Excel based.

1	A B	C D	E		F	G	н		1
1	Reco Jan 20	onciliation S	Statement					2023-08-08 6:32	EZ
3 4 5	Bank (	Closing Balance	•	\$	123.00				
6	In ledger	r, not in bank							
7		Adjustments		\$ (	597,742.99)				
8		Revised	Bank Balance	\$ (	597,619.99)				
9									
10	Ledge	r Closing Balan	ce	\$	123.00				
11	In bank,	not in ledger							
12		Deposits		\$	7,528.18				
13		Payments		\$	29,337.64				
14		Transfers		\$	37,124.18				
15			Adjustment:						
16		Revised Le	edger Balance	\$	74,113.00				
17			-						
18			Difference	\$10	571,732.99)				
19									
20									
21									

Above is the reconciliation statement report.

In Le	dger, Not In B	ank				2023-08-07 19-48	EZ
Pariod	Group Data	Journal id	Decembra	Durmont Mathed Durmont Def Mbr	Voodor Name	Hear Commonte	Amount
Feb 2023 og	Adjustments (18-31-22	EZADJE500277	ADJUSTMENT	Payment Method Payment Ref. Nor.	vendor mame	Oser Comments	\$ 27.50
Feb 2023 og	Adjustments 08-31-22	E7AD IE500278	ADJUSTMENT	0			\$ 27.50
Feb 2023 og	Adjustments 08-31-22	E7AD (E500279	ADJUSTMENT	0			\$ 27.50
Feb 2023 ng	Adjustments 08-31-22	EZAD.(E500280	ADJUSTMENT	0			\$ 20.00
Feb 2023 pg	Adjustments 08-31-22	E7AD (E500281	ADJUSTMENT	0			\$ 27.50
Feb 2023 og	Adjustments 08-31-22	E7AD (E500282	ADJUSTMENT	0			\$ (191.03
Feb 2023 og	Adjustments 08-31-22	E7ADJE500282	ADJUSTMENT	0			\$ 191.03
Feb 2023 og	Adjustments 08-31-22	E7AD (E500283	ADJUSTMENT	0			\$ 27.50
Feb 2023 og	Adjustments 08-03-22	EZADJE500284	ADJUSTMENT	0			\$ (9.00
Feb 2023 og	Adjustments 08-03-22	E7AD.IE600284	ADJUSTMENT	0			\$ (6.40
eb 2023 og	Adjustments 08-03-22	EZADJE500284	ADJUSTMENT	0			S (2.50
Feb 2023 og	Adjustments 08-02-22	EZAD.IE500305	ADJUSTMENT	0			\$ 90.17
Feb 2023 og	Adjustments 08-03-22	EZADJE500306	ADJUSTMENT	0			\$ 64.41
eb 2023 og	Adjustments 08-04-22	EZADJE500307	ADJUSTMENT	0			\$ 101.85
eb 2023 og	Adjustments 08-31-22	EZADJE500308	ADJUSTMENT	0			5 (0.10
Feb 2023 og	Adjustments 08-31-22	EZADJE500308	ADJUSTMENT	0			\$ 0.10
eb 2023 og	Adjustments 08-31-22	EZADJE500309	ADJUSTMENT	0			5 (0.02
eb 2023 og	Adjustments 08-31-22	EZADJE500318	ADJUSTMENT	0			\$ 27.50
Feb 2023 og	Adjustments 08-17-22	EZADJE500330	ADJUSTMENT	0			\$ 18.72
eb 2023 og	Adjustments 08-31-22	EZADJE500336	ADJUSTMENT	0			S 0 10
eb 2023 og	Adjustments 08-31-22	EZAD./E500336	ADJUSTMENT	0			S (0.10
eb 2023 og	Adjustments 08-02-22	EZADJE500390	ADJUSTMENT	1234			\$ (88.00
Adj By All SUM	ustments SUM						\$ (597.742.99 \$ (597.742.99

Above is a sample of the unreconciled items report.

Transaction Verification

		Bank				Ledger		
Period	Closing Balnce	Trans. Should Total	Trans. Summary	Result	Closing Balnce	Trans. Should Total	Trans. Summary	Result
Jan-23	\$420.00	\$420.00	\$66,820.18	-\$66,400.18	\$45.00	\$45.00	\$66,820.18	-\$66,775.18
Feb-23	\$123.00	-\$297.00	\$7,269.82	-\$7,566.82	\$123.00	\$78.00	\$7,269.82	-\$7,191.82
Mar-23	\$500.00	\$377.00	\$0.00	\$377.00	\$400.00	\$277.00	\$0.00	\$277.00
Apr-23	\$150.00	-\$350.00	\$0.00	-\$350.00	\$200.00	-\$200.00	\$0.00	-\$200.00

EZ

2023-08-19 9:50

Above is a sample of the Transaction Verification Report

The Historical Reports contains the list of reports that were generated and were saved in the system. Only super users can delete reports that were saved.

BUILT-IN REPORTS	Saved Report History				
HISTORICAL REPORTS					
USER REPORTS	File Name	Accounting Period	Date Saved	Saved By	Action
	Statement_2023_08_07_nbd_2023_7.xlsx	Jan 2023 _og	2023-08-07 9:33:23 PM	Kelvin Woodman	⊉ 📋

Only Reconciliation statement is available to save to the system.

The user reports contain any user generated report. These reports can be configured and they must be Sql Reporting Services.

#### Reporting

BUILT-IN REPORTS	Report Select Report	
USER REPORTS	DISPLAY REPORT 11.	

## Matched Transaction Details

Select the Matched Trans. Details menu option.

≡ National Ban	k — 8655588441	
Matched Tr	ansaction Det	ails
Transactions Matched Betwee 2023-07-28	→ 2023-07-30 Q SEARCH	ā
BANK TRANSACTIONS	LEDGER TRANSACTIONS	

This screen allows you to see the transaction which were matched between two given dates. Once the dates are entered, and the search button pressed, the system will list all transactions from both bank and ledger sides.

#### Matched Transaction Details

23-06-01	→ 2023-07-31			
	Q SEARCH			
IK TRANSACTIONS L	EDGER TRANSACTIONS			
LEDGER TRANSACTION	S Journal Id Y	Effective Date 🛛 🍸	Description Y	Payment Ref. No 🛛 🏹
LEDGER TRANSACTION Amount (\$0.10)	S Journal Id Y EZADJF500308	Effective Date 2022-08-31	Description Y ADJUSTMENT	Payment Ref. No 🛛 🏹
LEDGER TRANSACTION Amount Y (\$0.10) \$0.10	S Journal Id EZADJF500308 EZADJF500308	Effective Date	Description     ▼       ADJUSTMENT       ADJUSTMENT	Payment Ref. No ⊽ 0 0
LEDGER TRANSACTION Amount ♀ (\$0.10) \$0.10 \$0.10	S Journal Id ZADJF500308 EZADJF500308 EZADJF500336	Effective Date ▼ 2022-08-31 2022-08-31 2022-08-31	Description     Y       ADJUSTMENT     ADJUSTMENT       ADJUSTMENT     ADJUSTMENT	Payment Ref. No 0 0 0

The tab BANK TRANSACTIONS & LEDGER TRANSACTIONS allows you to see the respective transactions that are within the date range. Once a transaction is selected, the transactions which were matched with selected transaction is displayed at the bottom of the tab.

LEDGER T	RANSACTIONS				
Actions	Amount 🛛	Journal Id Y	Effective Date Y	Description Y	Payment Ref. No
<b>^</b>	(\$0.10)	EZADJF500308	2022-08-31	ADJUSTMENT	0
5	\$0.10	EZADJF500308	2022-08-31	ADJUSTMENT	0
<b>n</b>	\$0.10	EZADJF500336	2022-08-31	ADJUSTMENT	0
6	(\$0.10)	EZADJF500336	2022-08-31	ADJUSTMENT	0
ANK TRA	NSACTIONS MATCHED WIT	H LEDGER TRANSACTIONS	MATCHED WITH		
	ng V An	nount 🍸 Pos	sting Date 🛛 De	scription 1 Y	Description 2 Y

To un-match transactions. If your user is a Super User, you will see a button ( 🔊 ) next to the

transaction. Pressing this button will un-match the transaction and all other transactions matched along with that transaction if needed.

# Search For Transactions

This screen allows you to search for a transaction. The system searches the description fields of the transactions. All matching transactions are returned.

Search for 1	Fransactions
Search	۹
٩	SEARCH
BANK TRANSACTIONS	LEDGER TRANSACTIONS

Enter a term and press the search button. The system will return all matching transactions on both bank and ledger side irrespective of whether or not they are matched.

	ransactions			
Search				
adj	Q			
Q, SE	ARCH			
NIV TRANCACTIONS	EDGED TDANCACTIONS			
	LEDGER TRANSACTIONS			
LEDGER TRANSACTION	NS			
LEDGER TRANSACTION	Journal Id Y	Effective Date 7	Description Y	Payment Ref. No 🛛 🏹
LEDGER TRANSACTION Amount Y (\$65,000.00)	Journal Id Y ADJF5001067	Effective Date Y 2022-07-31	Description Y ADJUSTMENT	Payment Ref. No 🛛
LEDGER TRANSACTION Amount ♀ (\$65,000.00) (\$148,678.06)	NS Journal Id Y ADJF5001067 ADJF5001062	Effective Date ♀ 2022-07-31 2022-07-31	Description ADJUSTMENT ADJUSTMENT	Payment Ref. No 🖓 0 0
LEDGER TRANSACTION Amount ¥ (\$65,000.00) (\$148,678.06) \$100.00	NS Journal Id Y ADJF5001067 ADJF5001062 EZADJF500389	Effective Date ♀ 2022-07-31 2022-07-31 2022-07-26	Description     Y       ADJUSTMENT       ADJUSTMENT       ADJUSTMENT       ADJUSTMENT	Payment Ref. No      ∇     0     0     0     0
LEDGER TRANSACTION Amount 7 (\$65,000.00) (\$148,678.06) \$100.00 (\$28,075.00)	Journal Id         ▼           ADJF5001067         ADJF5001062           EZADJF500389         ADJF5001643	Effective Date         Y           2022-07-31         2022-07-31           2022-07-26         2022-07-31	Description     Y       ADJUSTMENT        ADJUSTMENT        ADJUSTMENT        ADJUSTMENT        ADJUSTMENT	Payment Ref. No      ∇     0     0     0     0     0     0

# **Staging Area**

Let's look at the Staging Menu. Before any transaction get's uploaded for processing, it goes to the staging area for "pre-processing"



There are four menu items in the staging menu, which are available only to super users.

- Closing Balances
- Import Transactions
- Bank Transactions
- Ledger Transactions

#### **Closing Balances**

This screen is used to maintain the closing balances for the bank account and the ledger account. Balances cannot be edited once entered. However, they can be deleted. These balances are necessary as they will allow for the reconciliation to be accurate.

≡ National Bank – 8	655588441				Kelvin Woodman 🔘 [+
Maintain Closin	ng Balances				
Closing Balances					ADD CLOSING BALANCE
Accounting Period	Year	Period	Bank	Ledger	
Jan 2023 _og	2023	7	\$123.00	\$123.00	DELETE
Feb 2023_og	2023	2	\$420.00	\$45.00	DELETE
			Rows per p	oage: 10 👻 1-2 of 2	IC C > >1

To add a new period closing balance, press the ADD CLOSING BALANCE button. You will be presented with the following screen:

Accounting Period	
Select accounting period	*
Bank Closing Balance*	
\$ 0	
Ledger Closing Balance*	6
\$ 0	(4)

Select the accounting period. Enter the figure on the bank statement. You may enter the figure from the ledger or press the blue arrow button to load the balance directly from Smartstream.

Once done press the ADD BALANCES button to save.

# Import Transactions

This screen allows you to upload the file from the bank, from the ledger program or directly from Smartstream. It supports the following file types XML,CSV,XLS,XLSX.

# Import Transactions

Accounting Period Select accounting period	*		
Bank Transactions O	LedgerTransactions		
Drag and drop files here o	or click		
he following file formats are supported: X	ML,CSV,XLS,XLSX		
LIPLOAD TO BANK STACING	CLEAD		

To begin, select the accounting period you are importing the transactions into.

Then select the type of transactions, whether bank or ledger. Note: if you select Ledger, you will have the option to import the transactions directly from Smartstream as shown below.



You may drag and drop the file into the drop zone or clock in the drop zone to open the file browser to select the file. Once the file is ready to be uploaded it will appear in the drop zone as follows and the buttons at the bottom of the screen will be enabled allowing you to upload the file or clear the file from the system.

Jan 2027				*		
Bank Tra	nsactions	0	LedgerTransac	tions		
Drog ond d	rop files h	nere	or click			
rag and d	-					
January 2, 20	23.csv					
January 2, 20	123.csv					
January 2, 20	123.csv					
January 2, 20	923.csv					

One the transactions are uploaded they are in the staging area, Bank Transactions and Ledger Transactions waiting for further processing.

#### Bank and Ledger Staging Screens

These two screens are identical in function. They allow you to double check and verify the transactions before it moves into the work area. It also allows you to group the transactions.

Note: The accounting period will only be enabled if there are multiple periods in the staging area. i.e. if you imported files from two different periods (eg. You imported files from January 2021, and February 2021)

#### Bank Transactions - Staging Area

IMPORT FOR RECOCILLIATION

There are totals at the bottom of the screen to assist with the verification of transactions and balances.

Accounting Period Select accounting	period	*				
🚀 RUN PROCES	SES	∎≓ DELE	TE STAGING TRANSACTIONS	4 IMPOR	FOR RECOCILLIATION	
Bank Transact	tions		All Groups			ш
Grouping	Amount	Posting Date	Description 1	Description 2	Description 3	
Select group 👻	(\$350.00)	2023-01-02	02301020000022 10280802 P.A.Y.E Tax	Receipt # 271070 dated 16.01.2023	PAYE Arrears obo International University for Graduate Studies	C
Select group 👻	\$1,000.00	2023-01-02	DEPOSIT	SOME	PIT 3	٦
Select group.	\$325.00	2023-01-02	DEPOSIT			F

The RUN PROCESSES button runs any custom process that may be necessary. For example, you may set up a process to convert the currency, or automatically set the grouping based on a field description.

Once your verification is complete, press the IMPORT FOR RECONCILIATION button. This will import the transactions for reconciliation. NOTE. Any transaction without a grouping will not be imported. Also, note that this button will be disabled if the sum of the imported transactions match the closing balance for the period.

If transactions were imported into the wrong period, you may delete the transactions by pressing the button "DELETE STAGING TRANSACTIONS".

Individual transactions can be deleted by using the delete icon button in the transaction row.

# Maintenance Area

Now we have the Maintenance menu. This menu option is only available to super users. It provides access to create the accounting periods, adjustments and view a partial history of what happened on the account.



The following menu options are present:

- Accounting Period
- SS Clearance File
- Adjustments
- View Process History

#### Accounting Period

The Accounting Period screen allows us to create new accounting periods for use in the system.

Periods	ADD /	ACCOUNTING	PERIOD AUTO-GI	EN ACCOUN	TING PERIOD		Q Search			
Period 🛛	Accounting Period	Ÿ	Accounting Year	Ŷ	SS Posting Period	Ŷ	SS Posting Year	Ŷ		
Jan 2027	1		2027		1		2023		EDIT	DELETE
Sep 2026	9		2026		9		2027		EDIT	DELETE
Feb 2026	2		2026		2		2026		EDIT	DELETE
Jan 2026	2		2026		1		2026		EDIT	DELETE
Sep 2023	9		2025		9		2026		EDIT	DELETE

#### Accounting Periods

To create a new period, press the ADD ACCOUNTING PERIOD button. The system will then open the Add New Period dialog window.

100010 0000	
Period Description*	
Account Period*	<u>^</u>
0	Ç
Accounting Year*	
0	0
SS Posting Period*	
0	^
CC Dorting Voart	*
0	^
0	~

Enter a description of the period, eg. Jan 2023. Enter the account period which could represent the month and also the accounting period year. The fields SS Posting Period and SS Posting Year represent the Smartstream posting period and year respectively. (Due to different fiscal periods these do not correspond to calendar months) Once all the information is entered press the ADD PERIOD button to add the period.

To edit or delete a period, press the corresponding buttons on the grid next to the period you would like to edit/delete.

Once accounting periods for a year is entered, the system can generate new periods using a year as a basis.

Press the AUTO-GEN ACCOUNTING PERIOD button. The following dialog will open.

Base year*	
2027	<u>^</u>
New Year*	·
2028	^
2028	Ç

Specify the year to be used as a template and the year for which the periods should be generated. Once done click the ADD PERIOD button and the accounting periods will be generated for the year specified.

#### **SS** Clearance File

This screen is used to generate a file that clears the payments on the Smartstream system. This may or may not be necessary depending on your use of Smartstream.

Kelvin Woodman 🕜 [+

# Smartstream Clearance File

GENERATE CLEARANCE FILE

Press the GENERATE CLEARANCE FILE button and the system will produce a file containing the transactions which were matched for clearance in the Smartstream system.

#### Adjustments

This screen allows you to add any adjustments that are necessary for the reconciliation.

#### Manage Adjustments

Adjustments				ADD ADJUSTMENT
Amount	Comment	Created By	Date Created	
\$125,663.00	Initial Balance when we drew a line in the sand	Kelvin Woodman	2023-07-31 6:38:36 AM	DELETE
		Rows per page:	10 👻 1-1 of 1 🛛 I <	< > >1

To add an adjustment, press the ADD ADJUSTMENT button. The following dialog will appear.

Add New Adjustment		×
Adjustment Amount* \$ 0		
Comment		
	CANCEL	ADD ADJUSTMENT

Enter the amount and a comment. Then press the ADD ADJUSTMENT button. Once pressed the adjustment will be registered on the system. Adjustments cannot be edited, only deleted. To delete an adjustment, press the DELETE button next to the adjustment you want to delete.

#### **View Process History**

This screen shows you the date and user who ran what the system considers significant processes, for example adding or removing adjustments.

#### History of Processes Ran **Process History** Additional Details Process Y Y Accounting Period γ Ran By Y Date Ran Y 2023-07-31 6:38:36 AM Uploaded Bank File/Transaction Kelvin Woodman Adjustment: \$125,663.00 2023-07-31 6:38:36 AM Removed Adjustment Kelvin Woodman Removed Adjustment Adjustment: \$121,313.00 Kelvin Woodman 2023-07-31 6:38:11 AM Generate SmartStream Clearance File Kelvin Woodman 2023-07-31 6:33:05 AM

# Administrator Area

This section covers the menu items in the administrative area. There are various settings which will seldom change during regular operations and should be set up once the account is setup or requirements change

ezBankRecon	$\equiv$ National Bank – 865	5588441
A Home	Bank Transactior	n Groups
Reconciliation -		
💭 Staging 👻	Transaction Grouping	
🚓 Maintainance 👻	Group Name	Debit or Credit
G Administration		
Bank Groups	Payments	Debit
Ledger Groups	Transfers	Debit
P* Custom Processes		
료 Custom Reports	Deposits	Credit
🖧 Users & Access		
Bank Accounts		

The menu items are as follows:

- Bank Groups
- Ledger Groups
- Custom Processes
- Custom Reports
- Users & Access
- Bank Accounts

#### Bank and Ledger Groups

The Bank and Ledger Group screens allows you to create the classification that can be used for the Bank and ledger transactions respectfully. These classifications will be required for reporting and other processes. It is best to decide on the classification before you begin using the system. The system allows you to go as detailed or as generic as you would like. For example, you may just have debits and credits, or break them further into checks, bank transfers, adjustments etc.

Bank Transact	ion Groups		
Transaction Grouping		ADD GRO	DUP
Group Name	Debit or Credit		
Payments	Debit	EDIT	
Transfers	Debit	EDIT DELETE	
Deposits	Credit	EDIT DELETE	
		Rows per page: 10 ▼ 1-3 of 3 I< < >	>1

# Ledger Transaction Groups

Ledger Transaction Grouping		ADD GROUP
Group Name	Debit or Credit	
Cheques	Debit	EDIT
Bank Transfers	Debit	EDIT
Deposits	Credit	EDIT
Adjustments	Debit	EDIT
Test	Credit	EDIT
	Ro	ows per page: 10 ▼ 1-5 of 5 I< < > >I

To add a new group, press the ADD GROUP button. The following dialogue will open.

Add New Group		×
Description*		
O Debit O Credit	CANCEL	ADD GROUP

Enter the description of the group that you would like to create and select whether it goes in the debit or credit category. Click the ADD GROUP Button to save the new group.

Note. Your administrator may have to configure the stored procedures

## **Custom Processes**

The system allows you to create stored packages to run on your data. For example, you may need a function to split the description given by the bank. This screen allows you to add the function/procedure so that the end user can run the function after it is created.

Manage Custom F	Processes		
Custom Process			ADD CUSTOM PROCESS
Process Name	Stored Procedure		
Run Bank 1	bank1		DELETE
		Rows per page: 10 💌 1	-1 of 1 I < < > >I

To add a custom process, first make sure that it is added to the database under the correct account schema. Next use this screen to create a link to run the process. Click the ADD CUSTOM PROCESS button and the following dialog will open.

Enter a friendly name for the end user, and enter the actual name of the stored procedure. Hit the ADD CUSTOM PROCESS button to save. The custom process will now show up as an option on the Match Transactions screen.

**Custom Reports** 

The system has the ability to link to a Microsoft SQL Server Reporting Service. Therefore, any reports that are needed can be added to the system. For example, there may be a need for custom branding or report that pulls data from outside the system.

#### Manage Customs Reports

Custom Rep	orts		ADD CUSTOM REPORT
Display Name	Report Name	URL Base Address	Report Setting
Bank Budget	brh	https://myreportserver.com/	
LedgerAll	all	http://google.com	&rsCommand=Render&rs:Embed=true&rc:Toolbar=true&rc:Parameters=Collapsed
			Rows per page: 10 ▼ 1-2 of 2  < < > >

To enter a custom report, access the custom reports screen. Press the ADD CUSTOM REPORT button whereby the following dialog will open.

dd Nev	v Custom Report	)
Friendly	Name*	
Report N	iame*	
Base Ad	dress*	
URL Rewp	ort Setting	
Stored F	Procedure sername	
) A	ccount	
<b>A</b>	ccounting Period	
В	ank or Ledger	
B	ank Groups	
. D L	edger Groups	

Enter the information and then press the ADD REPORT button. The report will now be available for selection in the Reporting section.

Field	Description
Friendly Name	The name displayed for the user
Report Name	The actual name of the report
Base Address	The url of the report server
Url Report Settings	The tail portion of the url. Usually
	standara but can be customized
Stored Procedure	The name of the stored procedure
	to run before running the report

User Name	Pass the username to the report or not
Account	Pass the account to the report or not
Accounting Period	Display the accounting period selector
Bank or Ledger	Display the bank or ledger selector
Bank Group	Show the bank group selector
Ledger Group	Show the ledger group selector
Date Range	Show the date range selector

To Edit or Delete a report, press the corresponding button to the far right of the grid next to the report you would like to edit or delete.

# Manage Customs Reports

Custom Report	S					ADD CUSTOM REPORT
Account	Date Range	Bank or Ledger	Bank Grouping	Ledger Grouping	Period	
False	False	False	False	False	False	EDIT
True	True	True	True	True	True	EDIT
				Rows per page: 1	0 ▼ 1-2 of 2	ic c > >i

#### Users and Access

This screen is used to manage access and security of the system. Users of the system need to have an active directory account before they are able to use the software. Once an active directory account is created, you can proceed to add the user to the system.

# Manage Users and Access

Users		ADD NEW USER		Q Search	h			
id	Username	Name	Role(s)					
1	woodmank	Kelvin Woodman	Administrator Super User	Regular User		ROLES	ACCOUNTS	DELETE
2006	sb	Kelvin-Scotia	Regular User			ROLES	ACCOUNTS	DELETE
4006	Kelvin2	Kelvin2	Regular User			ROLES	ACCOUNTS	DELETE
				Rows per page:	10 🔻	1-3 of 3	< <	> >

To add a new user, press the ADD NEW USER button. The following dialog will open

Add New User		×
Name*		
Username* Enter the GOCD username		
	CANCEL	ADD USER

Enter the name of the user and the active directory user name. Click the ADD USER Button to add the user. Once the user is added, you now need to grant the user a role. The roles on the system are:

Role	Description
Regular user	Has access to Reconciliation Area
Super User	Has access to Staging Area
Administrator	Has access to Administrative Area

Press the ROLE button next to the user to grant the user a role. Select the role or roles that you would like to grant the user. Then click the SAVE button.

🗸 Regular User	🗸 Super User	✓ Administrator	

After roles are selected, the user will need to be granted access to an account. To add a user to an account select the ACCOUNTS button next to the user. The following dialog will open.

Account	Enable	Super
Account	Account	User
National Bank -	-	-
8655588441		
Scotia Bank - A3-77892		

The dialog will list all available accounts. Enable the account that you would like to give the user access to. If the user is a super user for the account, enable the super user field. Click the SAVE button when done.

#### **Bank Accounts**

The Bank Account screen maintains the different accounts that the system will reconcile.

Accounts		ADD NEW ACCOUNT		Q Search		
lame	Account No.	SS Account No.	Schema			
lational Bank	8655588441	12211	nbd	BANK FILE MAPPING	LEDGER FILE MAPPING	DELET
cotia Bank	A3-77892	212121	bns	BANK FILE MAPPING	LEDGER FILE MAPPING	DELET

Click the ADD NEW ACCOUNT button to add a new account to reconcile to the system. Once, he button is pressed the following dialog box will open.

Add New Account		>
Name*		
Account No.*		
Smartstream Account No.*		
Account Schema*		0.15
		0.7.5
	CANCEL	ADD ACCOUNT

Enter the account Name, number, Smartstream account number (this allows the system to pull information from the Smartstream system) and a unique combination for the schema. The schema is the database schema on which all the tables and procedures will be created under.

Once done press the ADD ACCOUNT button. The new account will be created along with the database tables.

The next matter to configure with the account is the Bank File Mapping and Ledger File Mapping. This maps the corresponding columns on the to the fields when doing file uploads into the system.

esponding colum	m number in the file for this	s field
~	Account No.*	<u>^</u>
~	3	~
0		
Ŷ	Currency* 5	Ŷ
	Description 4*	
^	9	^
~	Depositorion 5t	~
~	10	^
~	10	~
	Description 6*	
0	11	0
		Account No.* 3 Account No.* 3 Currency* 5 Currency* 5 Description 4* 9 Description 5* 10 Description 6* 11

You will not need to change anything here unless the file format from the bank has changed. If the file format from the Bank has changed, simply enter the corresponding column number with the corresponding field. Then press the save button.

edger File Mappi	ng		
Enter the corre	esponding colum	m number in the file for this f	ield
Journal Id*		Effective Date*	
1	0	3	0
Ledger Id*		Sequence No*	
0	^	A	^
6	~	-	~
Ministry*		Posting year*	
5	^	12	^
	~		~
Program*		Posting Period*	
6	0	13	0
Sub-Program*		Debit/Credit Code*	
7	^	14	^
,	~	11	~
Account*		Amount*	
8	\$	15	0
Project*		Date Posted*	
9	^	16	^
	~		~
SOF*		Payment Method*	3.2
10	^	17	^

The same goes for the Ledger file mapping. You would rarely, if ever need to use ledger file upload as the system should pull the information straight from the Smartstream system. Make sure that the column number matches the corresponding column.

If an account is no longer needed it may be deleted by pressing the DELETE button next to the account. Once the delete button is pressed a confirmation dialogue is presented to the user. If you choose to continue, you will also have to decide if you would like to keep the database objects that were created or not.

# **Advanced Configuration**

#### Database Overview

The database for this program changes with every account added or removed to the system. Tables with the "dbo" prefix are used by the main program. Any account specific tables are prefix by the "schema" value entered when the account was created.

The main tables are:

- [dbo].[account\_bank\_file\_mapping]
- [dbo].[account\_ledger\_file\_mapping]
- [dbo].[accounts]
- [dbo].[processes]
- [dbo].[roles]
- [dbo].[user\_accounts]
- [dbo].[user\_preferences]
- [dbo].[user\_roles]
- [dbo].[users]

## Account Related Tables

Each account has tables, views and procedures associated with it. The account tables is as follows:

- {schema}.[accounting\_periods]
- {schema}.[adjustments]
- {schema}.[bank\_transaction\_groups]
- {schema}.[bank\_transactions]
- {schema}.[bank\_transactions\_staging]
- {schema}.[closing\_balances]
- {schema}.[custom\_processes]
- {schema}.[custom\_reports]
- {schema}.[ledger\_transaction\_groups]
- {schema}.[ledger\_transactions]
- {schema}.[ledger\_transactions\_staging]
- {schema}.[process\_logs]
- {schema}.[reports\_history]
- {schema}.[user\_bank\_columns]
- {schema}.[user\_ledger\_columns]

#### **Stored Procedures**

Each account schema also contains the following stored procedures:

- {schema}.[bank\_staging\_process]
- {schema}.[generate\_ss\_flLe]
- {schema}.[import\_ledger\_transactions]
- {schema}.[ledger\_staging\_process]
- {schema}.[matching\_process]

Also 3 database view:

- {schema}.[ledger\_closing\_balance\_vw]
- {schema}.[ledger\_trans\_sum\_vw]
- {schema}.[ledger\_trans\_sum\_vw]

#### **Application Settings**

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
 },
  "AllowedHosts": "*"
  "ConnectionStrings": {
    "Default": "Data Source=WOODZONE; Initial Catalog=ezBankRecon; Integrated
Security=True;TrustServerCertificate=true",
    "Domain": "gocd.mofpdomgov.gov"
 },
 "BaseUrl":"/",
  "SOLLinkServer": "gocdssp",
  "TimeOut": "1200"
}
```

Setting	Description
Default	The connection string to the database
Domain	The domain that hat the users authenticate with
SQLLinkServer	The name of the link server created on the database that links to the smartstream server.
BaseUrl	The base url of the application. This is needed if application runs from a sub-domain

# **TECHNICAL INFORMATION**

# Technologies & Resources

#### .Net Blazor

The web program was built using the latest C# and Blazor web technologies. C# is a **modern**, **innovative**, **open-source**, **cross-platform** object-oriented programming language. And is used from enterprise development.

https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor

#### MudBlazor

Mudblazor is an opensource UI framework. It provides the various components which provide a consistent experience for the end user. It follows the Google Material Design system.

https://mudblazor.com/

#### Closedxml

ClosedXML is a .NET library for reading, manipulating and writing Excel 2007+ (.xlsx, .xlsm) files. It is used in the program to generate excel files.

https://docs.closedxml.io/en/latest/

#### Closedxml.Reports

ClosedXML.Report is a tool for report generation and data analysis in .NET applications through the use of Microsoft Excel. It is used in the program in the generation of standard reports.

https://closedxml.io/ClosedXML.Report/

#### Dapper

Dapper is a simple object mapper for .NET. It is used in the program to connect to the SQL Server database.

https://dapperlib.github.io/Dapper/

#### Signal R

SignalR allows server side code to push updates to connected clients (primarily over WebSockets) as soon as new content/information is available, without any need for clients to constantly poll the server for updates. It is used in the program to update and sync the transactions across different sessions for the users.

https://dotnet.microsoft.com/en-us/apps/aspnet/signalr

## Excel Data Reader

This is a lightweight and fast library written in C# for reading Microsoft Excel files. It is used in the program for processing the uploaded files.

https://github.com/ExcelDataReader/ExcelDataReader

#### Microsoft SQL Server

Microsoft SQL Server is used to power the back end database in the application.

https://www.microsoft.com/en-us/sql-server

# Installation

#### Requirements

The program runs on Microsoft Technologies. Due to .Net Core, it can run standalone if so desired. However, the supplied compile version was compiled to run on Microsoft ISS on windows sever.

Before you install on the windows server, please ensure that the .Net SDK is installed on the server.

You can download the SDK at

https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/sdk-7.0.305-windows-x64-installer

For more information please check out the following website.

https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/?view=aspnetcore-7.0#iis-configuration

Once the SDK is installed, you may use the WEB Deploy functionality to install the software. The software may be installed on it's own domain or part of a domain.

The application uses https. This provides secure communication. Therefore, a SSL certificate needs to be installed. The program will work without an SSL certificate. You may create a self signed certificate or get a one from a certificate authority.

#### Sample Installation

1. Open IIS



2. Select the site you would like to deploy the application to. Right Click. Select Deploy and then Import Application

File View Help	i.													
Connections			0-	Decena								Act	tions	
			GOC	DSSP2	Home								Manage Server	
Start Page				Group by: A	Area •				2	🕏 Restart				
GOCDSSP2 (G	OCD	woodmar	ASP NET		10.00	1000 C		55. 	Term		• •	₽	Start	
v Sites	n Pot	//5	- Ba			0	0			illio	(2000)		Stop	_
> 😔 Defaul	Mak	Site			404			y	ab	<b>N</b>			View Application Pools	
	28	Edit Darmi	reiner (	.NET moilation	NET Error	.NET Globalization	.NET Trust	Application	Connection	Machine Key	Pages and Controls		VIEW SIDES	
		Edit Permi	5510115									De	ploy	6
	1	Add Appli	cation	2	sion State SMTP E-mail							8	Export Server Package	
	8	Add Virtua	al Directory	sion State								•	Import server or site Par	xag
		Edit Bindir	ngs										Version	¥.
		Manage W	/ebsite +		-	-		-	1	-		0	Get New Web Platform	
	-	Refresh		2		THO	404			1		0	Components	_
	×	Remove		npression	npression Default Directory Error Page Document Browsing Configure Web Denloy Publishing		Error Pages	s Failed Handler HTTP HTTP		HTTP	- Help	rieip		
		Deploy	,	Co			ing.	Request Tra Mappings Redirect Respon				Respon		
	_	Penning		SI Exp	ort Applicatio	on		<u></u>		8				
		Nename		Import Application		e	s Modules Output Request S		Server					
Switch to Content View		× Del	Pelete Web Site and Content		Caching	aching Filtering	Certificates							
			~	• ve	ete vieb site	and content								

3. Follow the instructions on the wizard

Import Application Package

Specify the location of the package (fo Package path:	or example, C	\WebApplicat	ion.zip).		
C:\Backup\ezBankRecon.zip					Browse

#### 4. Once complete the application should be installed



?

5. Next, locate the config file in the inetpub, wwwroot directory. Open the folder, ezrecon and edit the appsettings.json file.



6. Make sure that the settings under the "ConnectionStrings" are pointing to the correct domain and correct database. And that the SQLLinkServer is configured if you have the databased linked. Save the settings and run the open the website.



7. Navigate to the {url}/admin/setup to configure the database. The setup installs the database tables and creates a new admin user.

	÷J
Setup New Database	
Admin Name*	
Admin Username*	
SETUP DATABASE	

Enter the admin information and click SETUP DATABASE.

Once done you will get a confirmation. The program is now ready to be used.

# Database Schema

ble Name Purpose
mapping] Used to store the bank file mappings for the columns for when the application uploads a file. This is stored per
account
mapping] Used to store the ledger file mappings for the columns for when the application uploads a file. This is stored per account
accounts] This stores the various accounts that are entered in the application that are available for reconciliation
processes] A list of standard processes that the system tracks
o].[roles] Stores a description of the roles in the database
accounts] Stores the accounts that a user is able to access on the system
eferences] Stores the user preferences
ser_roles] Stores the roles assigned to the users
o].[users] Stores the users who are able to access the system
mapping]Used to store the ledger file mappings for the columns for when the application uploads a file. This is stored per accounts"accounts]This stores the various accounts that are entered in the application that are available for reconciliation A list of standard processes that the system trackspo].[roles]Stores a description of the roles in the database systemaccounts]Stores the accounts that a user is able to access on the systemeferences]Stores the user preferencesser_roles]Stores the roles assigned to the userspo].[users]Stores the users who are able to access the system

The following tables are generated per each account

{schema}.[accounting_periods]	Stores the accounting periods
{schema}.[adjustments]	Stores the adjustments that are entered
{schema}.[bank_transaction_groups]	Stores the groupings decided on for the bank transactions
{schema}.[bank_transactions]	Stores the bank transactions that are used in the reconciliation process
{schema}.[bank_transactions_staging]	Houses transactions that are uploaded until they are ready for processing. This is where the transactions are uploaded to
{schema}.[closing_balances]	Stores the closing balances for the various periods
{schema}.[custom_processes]	Store the list of custom processes that are available for the end user to run
{schema}.[custom_reports]	Stores a list of reports for the end user to run
{schema}.[ledger_transaction_groups]	Stores the groupings decided on for the ledger transactions
{schema}.[ledger_transactions]	Stores the ledger transactions that are used in the reconciliation process
{schema}.[ledger_transactions_staging]	Houses transactions that are uploaded until they are ready for processing. This is where the transactions are uploaded to
{schema}.[process_logs]	Stores partial audits of action taken in the system
{schema}.[reports_history]	Stores the saved reconciliation reports
{schema}.[user_bank_columns]	Stroes the user preference for the columns displayed in the bank table
{schema}.[user_ledger_columns]	Stroes the user preference for the columns displayed in the ledger table

# APPENDIX 1 – Scripts for Main Tables

account\_bank\_file\_mapping

```
IF OBJECT_ID(N'[dbo].[account_bank_file_mapping]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[account_bank_file_mapping](
       [account_id] [int] NOT NULL,
       [branch] [int] NOT NULL,
       [customer] [int] NOT NULL,
       [account_no] [int] NOT NULL,
       [amount] [int] NOT NULL,
       [currency] [int] NOT NULL,
       [descp1] [int] NOT NULL,
      [descp2] [int] NOT NULL,
[descp3] [int] NOT NULL,
       [descp4] [int] NOT NULL,
       [descp5] [int] NOT NULL,
       [descp6] [int] NOT NULL,
       [posting_date] [int] NOT NULL,
       [value_date] [int] NOT NULL,
       [clearing_date] [int] NOT NULL,
       [trans_no] [int] NOT NULL,
       [posting_no] [int] NOT NULL,
 CONSTRAINT [PK_bank_file_mapping] PRIMARY KEY CLUSTERED
(
       [account_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END
```

#### account\_ledger\_file\_mapping

```
IF OBJECT_ID(N'[dbo].[account_ledger_file_mapping]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[account_ledger_file_mapping](
      [account_id] [int] NOT NULL,
      [jrnl_id] [int] NOT NULL,
      [ldr_entity_id] [int] NOT NULL,
      [eff_date] [int] NOT NULL,
      [jrnl_seq_nbr] [int] NOT NULL,
      [ministry] [int] NOT NULL,
      [program] [int] NOT NULL,
      [subprog] [int] NOT NULL,
      [account] [int] NOT NULL,
      [project] [int] NOT NULL,
      [sof] [int] NOT NULL,
      [sector] [int] NOT NULL,
      [posting_yr] [int] NOT NULL,
      [posting_pd] [int] NOT NULL,
      [dr_cr_code_1] [int] NOT NULL,
      [descp] [int] NOT NULL,
      [trans_amt] [int] NOT NULL,
      [date_posted] [int] NOT NULL,
      [pmt_meth_id] [int] NOT NULL,
      [pmt_ref_nbr] [int] NOT NULL,
```

```
[pmt_ref_date] [int] NOT NULL,
      [bank_id] [int] NOT NULL,
      [bank_acct_nbr] [int] NOT NULL,
      [vendor_id] [int] NOT NULL,
      [vendor_loc_code] [int] NOT NULL,
      [vendor_name] [int] NOT NULL,
      [payable_entity_id] [int] NOT NULL,
      [pmt_rqst_gross_amt] [int] NOT NULL,
      [pmt_amt] [int] NOT NULL,
      [type_of_pmt] [int] NOT NULL,
CONSTRAINT [PK_account_ledger_file_mapping] PRIMARY KEY CLUSTERED
(
      [account_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;",
@"IF OBJECT_ID(N'[dbo].[accounts]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[accounts](
      [id] [int] IDENTITY(1,1) NOT NULL,
      [account_name] [varchar](100) NOT NULL,
      [account_schema] [varchar](5) NOT NULL,
      [account_no] [varchar](50) NOT NULL,
      [ss_account_no] [varchar](50) NOT NULL
CONSTRAINT [PK_ez.accounts] PRIMARY KEY CLUSTERED
(
      [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [UK_account_schema] UNIQUE NONCLUSTERED
(
      [account_schema] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;
                                     processes
IF OBJECT_ID(N'[dbo].[processes]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[processes](
      [id] [int] NOT NULL,
      [process] [varchar](50) NOT NULL,
CONSTRAINT [PK_ez.process] PRIMARY KEY CLUSTERED
(
      [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
  INSERT INTO [dbo].[processes]([id],[process])
     VALUES (1, 'Added Closing Balance'), (2, 'Removed Closing Balance'), (3, 'Imported
Bank Transaction'),(4,'Imported Ledger Transactions'),(5,'Uploaded Bank
File/Transaction'), (6, 'Uploaded Ledger File/Transactions'), (7, 'Ledger Staging
Process'), (8, 'Bank Staging Process'), (9, 'Added Adjustment'), (10, 'Removed
```

Adjustment'),(11,'Ran Auto Matching Process'),(12,'Ran Custom
Process'),(13,'Generate SmartStream Clearance File');

END;

```
roles
```

END;

#### user\_accounts

```
IF OBJECT_ID(N'[dbo].[user_accounts]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[user_accounts](
      [user_id] [int] NOT NULL,
      [account_id] [int] NOT NULL,
      [enable_super] [bit] NOT NULL
) ON [PRIMARY];
END;",
@"IF OBJECT_ID(N'[dbo].[user_preferences]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[user_preferences](
      [user_id] [int] NOT NULL,
      [grid_size] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
      [user_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;
```

#### users

IF OBJECT\_ID(N'[dbo].[users]', N'U') IS NULL BEGIN
 CREATE TABLE [dbo].[users](
 [id] [int] IDENTITY(1,1) NOT NULL,
 [name] [varchar](50) NOT NULL,
 [username] [varchar](50) NOT NULL,

```
CONSTRAINT [PK_ez.users] PRIMARY KEY CLUSTERED
(
      [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [UK_users] UNIQUE NONCLUSTERED
(
      [username] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;",
@"IF OBJECT_ID(N'[dbo].[user_roles]', N'U') IS NULL BEGIN
      CREATE TABLE [dbo].[user_roles](
      [user_id] [int] NOT NULL,
[role_id] [int] NOT NULL,
CONSTRAINT [PK_ez.user_roles] PRIMARY KEY CLUSTERED
(
      [user_id] ASC,
      [role_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;"
@"INSERT INTO [dbo].[users]([name],[username])
     VALUES('{0}','{1}');",
@"INSERT INTO [dbo].[user_roles]([user_id],[role_id])
     VALUES(1,1);
```

# APPENDIX 2 – Scripts for Account Tables

```
accounting_periods
 IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = '{0}')
                   BEGIN
                    EXEC( 'CREATE SCHEMA {0}' );
                   END;"
            ,@"IF OBJECT_ID(N'{0}.[accounting_periods]', N'U') IS NULL BEGIN
      CREATE TABLE {0}.[accounting_periods](
             [id] [int] IDENTITY(1,1) NOT NULL,
             [descp] [varchar](50) NOT NULL,
             [posting_pd] [int] NOT NULL,
             [posting_year] [int] NOT NULL,
             [accounting_pd] [int] NOT NULL
             [accounting_year] [int] NOT NULL,
       CONSTRAINT [PK_{0}_accounting_period] PRIMARY KEY CLUSTERED
      (
             [id] ASC
      )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
      ) ON [PRIMARY];
END; ",
@"IF OBJECT_ID(N'{0}.[adjustments]', N'U') IS NULL BEGIN
      CREATE TABLE {0}.[adjustments](
             [id] [int] IDENTITY(1,1) NOT NULL,
             [amount] [money] NOT NULL,
             [comment] [varchar](500) NULL,
             [entered_by] [int] NULL,
             [date_imported] [datetime] NULL
      ) ON [PRIMARY];
```

```
END
```

#### bank\_transaction\_groups

```
IF OBJECT_ID(N'{0}.[bank_transaction_groups]', N'U') IS NULL BEGIN
      CREATE TABLE {0}.[bank_transaction_groups](
             [id] [int] IDENTITY(1,1) NOT NULL,
             [descp] [varchar](50) NOT NULL,
             [is_debit] [bit] NOT NULL,
       CONSTRAINT [PK_{0}_bank_code_groups] PRIMARY KEY CLUSTERED
      (
             [id] ASC
      )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
      ) ON [PRIMARY];
INSERT INTO {0}.[bank_transaction_groups](
             [descp],
             [is_debit])
VALUES('Credit',0),('Debit',1);
END;
```

#### bank\_transactions

IF OBJECT\_ID(N'{0}.[bank\_transactions]', N'U') IS NULL BEGIN CREATE TABLE {0}.[bank\_transactions]( [id] [int] IDENTITY(1,1) NOT NULL, [branch] [varchar](50) NULL, [customer] [varchar](50) NULL, [account\_no] [varchar](50) NULL, [amount] [money] NULL, [currency] [varchar](10) NULL, [descp1] [varchar](500) NULL, [descp2] [varchar](500) NULL, [descp3] [varchar](500) NULL, [descp4] [varchar](500) NULL, [descp5] [varchar](500) NULL, [descp6] [varchar](500) NULL, [posting\_date] [date] NULL, [value\_date] [date] NULL, [clearing\_date] [date] NULL, [trans\_no] [int] NULL, [posting\_no] [int] NULL [accounting\_period\_id] [int] NULL, [matched] [bit] NOT NULL, [matched\_bank\_transactions] [varchar](max) NULL, [matched\_ledger\_transactions] [varchar](max) NULL, [date\_matched] [datetime] NULL, [matched\_by] [int] NULL, [user\_comments] [varchar](max) NULL, [row\_color] [varchar](15) NULL, [is\_locked] [bit] NOT NULL, [lock\_user] [int] NULL, [date\_imported] [datetime] NULL, [import\_user] [int] NULL, [bank\_transaction\_group\_id] [int] NOT NULL, CONSTRAINT [PK\_{0}\_bank\_transactions] PRIMARY KEY CLUSTERED ( [id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY];

ALTER TABLE {0}.[bank\_transactions] CHECK CONSTRAINT [FK\_{0}\_bank\_transactions\_accounting\_periods];

END;

#### bank\_transactions\_staging

IF OBJECT\_ID(N'{0}.[bank\_transactions\_staging]', N'U') IS NULL BEGIN
 CREATE TABLE {0}.[bank\_transactions\_staging](
 [id] [int] IDENTITY(1,1) NOT NULL,
 [branch] [varchar](50) NOT NULL,

[customer] [varchar](50) NOT NULL, [account\_no] [varchar](50) NOT NULL, [amount] [money] NOT NULL, [currency] [varchar](10) NOT NULL, [descp1] [varchar](500) NULL, [descp2] [varchar](500) NULL, [descp3] [varchar](500) NULL, [descp4] [varchar](500) NULL, [descp5] [varchar](500) NULL, [descp6] [varchar](500) NULL, [posting\_date] [date] NULL, [value\_date] [date] NULL, [clearing\_date] [date] NULL, [trans\_no] [int] NULL, [posting\_no] [int] NULL [accounting\_period\_id] [int] NULL, [matched] [bit] NOT NULL, [matched\_bank\_transactions] [varchar](max) NULL, [matched\_ledger\_transactions] [varchar](max) NULL, [date\_matched] [datetime] NULL, [matched\_by] [int] NULL, [user\_comments] [varchar](max) NULL, [row\_color] [varchar](15) NULL, [is\_locked] [bit] NOT NULL, [lock\_user] [int] NULL, [date\_imported] [datetime] NULL, [import\_user] [int] NULL, [bank\_transaction\_group\_id] [int] NULL, CONSTRAINT [PK\_{0}\_bank\_transactions\_staging] PRIMARY KEY CLUSTERED ( [id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY];

ALTER TABLE {0}.[bank\_transactions\_staging] WITH CHECK ADD CONSTRAINT [FK\_{0}\_bank\_transactions\_staging\_accounting\_periods] FOREIGN KEY([accounting\_period\_id]) REFERENCES {0}.[accounting\_periods] ([id]);

ALTER TABLE {0}.[bank\_transactions\_staging] CHECK CONSTRAINT [FK\_{0}\_bank\_transactions\_staging\_accounting\_periods];

END;

#### closing\_balances

IF OBJECT\_ID(N'{0}.[closing\_balances]', N'U') IS NULL BEGIN CREATE TABLE {0}.[closing\_balances]( [id] [int] IDENTITY(1,1) NOT NULL, [bank\_balance] [money] NOT NULL, [accounting\_period\_id] [int] NOT NULL, [ledger\_balance] [money] NOT NULL, [entered\_by] [int] NOT NULL,

```
[date_entered] [datetime] NOT NULL,
CONSTRAINT [PK_{0}_closing_balances] PRIMARY KEY CLUSTERED
(
       [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
END;
```

#### ledger\_transaction\_groups

```
IF OBJECT_ID(N'{0}.[ledger_transaction_groups]', N'U') IS NULL BEGIN
CREATE TABLE {0}.[ledger_transaction_groups](
        [id] [int] IDENTITY(1,1) NOT NULL,
        [descp] [varchar](50) NOT NULL,
        [is_debit] [bit] NOT NULL,
        CONSTRAINT [PK_{0}_ss_code_groups] PRIMARY KEY CLUSTERED
        (
        [id] ASC
        )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
        ) ON [PRIMARY];
INSERT INTO {0}.[ledger_transaction_groups](
        [descp],
        [is_debit])
```

VALUES('Credit',0),('Debit',1);

END;

#### ledger\_transactions

```
IF OBJECT_ID(N'{0}.[ledger_transactions]', N'U') IS NULL BEGIN
      CREATE TABLE {0}.[ledger_transactions](
             [id] [int] IDENTITY(1,1) NOT NULL,
             [jrnl_id] [varchar](50) NOT NULL,
             [ldr_entity_id] [nchar](10) NOT NULL,
             [eff_date] [date] NOT NULL,
             [jrnl_seq_nbr] [int] NOT NULL,
             [ministry] [nchar](10) NOT NULL,
             [program] [nchar](10) NOT NULL,
             [subprog] [nchar](10) NOT NULL,
             [account] [nchar](10) NOT NULL,
             [project] [nchar](10) NOT NULL,
             [sof] [nchar](10) NOT NULL,
             [sector] [nchar](10) NOT NULL,
             [posting_yr] [int] NOT NULL,
             [posting_pd] [int] NOT NULL,
             [dr_cr_code_1] [nchar](10) NULL,
             [descp] [varchar](100) NULL,
             [trans_amt] [money] NOT NULL,
             [date_posted] [date] NOT NULL,
             [pmt_meth_id] [nchar](10) NULL,
             [pmt_ref_nbr] [int] NULL,
             [pmt_ref_date] [date] NULL,
```

[bank\_id] [nchar](10) NULL, [bank\_acct\_nbr] [varchar](50) NULL, [vendor\_id] [varchar](50) NULL, [vendor\_loc\_code] [nchar](10) NULL, [vendor\_name] [varchar](50) NULL, [payable\_entity\_id] [nchar](10) NULL, [pmt\_rqst\_gross\_amt] [money] NULL, [pmt\_amt] [money] NULL, [type\_of\_pmt] [nchar](10) NULL, [accounting\_period\_id] [int] NULL, [matched] [bit] NOT NULL, [matched\_bank\_transactions] [varchar](max) NULL, [matched\_ledger\_transactions] [varchar](max) NULL, [date\_matched] [datetime] NULL, [matched\_by] [int] NULL, [user\_comments] [varchar](max) NULL, [row\_color] [varchar](15) NULL, [is\_locked] [bit] NOT NULL, [lock\_user] [int] NULL, [date\_imported] [datetime] NULL, [import\_user] [int] NULL, [ledger\_transaction\_group\_id] [int] NULL, [generated] [bit] NULL, [date\_generated] [datetime] NULL, CONSTRAINT [PK\_{0}\_ledger\_transactions] PRIMARY KEY CLUSTERED ( [id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY];

ALTER TABLE {0}.[ledger\_transactions] CHECK CONSTRAINT [FK\_{0}\_ledger\_transactions\_accounting\_periods];

END;

#### ledger\_transactions\_staging

[posting\_pd] [int] NOT NULL, [dr\_cr\_code\_1] [nchar](10) NULL, [descp] [varchar](100) NULL, [trans\_amt] [money] NOT NULL, [date\_posted] [date] NOT NULL, [pmt\_meth\_id] [nchar](10) NULL, [pmt\_ref\_nbr] [int] NULL, [pmt\_ref\_date] [date] NULL, [bank\_id] [nchar](10) NULL, [bank\_acct\_nbr] [varchar](50) NULL, [vendor\_id] [varchar](50) NULL, [vendor\_loc\_code] [nchar](10) NULL, [vendor\_name] [varchar](50) NULL, [payable\_entity\_id] [nchar](10) NULL, [pmt\_rqst\_gross\_amt] [money] NULL, [pmt\_amt] [money] NULL, [type\_of\_pmt] [nchar](10) NULL, [accounting\_period\_id] [int] NULL, [matched] [bit] NOT NULL, [matched\_bank\_transactions] [varchar](max) NULL, [matched\_ledger\_transactions] [varchar](max) NULL, [date\_matched] [datetime] NULL, [matched\_by] [int] NULL, [user\_comments] [varchar](max) NULL, [row\_color] [varchar](15) NULL, [is\_locked] [bit] NOT NULL, [lock\_user] [int] NULL, [date\_imported] [datetime] NULL, [import\_user] [int] NULL, [ledger\_transaction\_group\_id] [int] NULL, [generated] [bit] NULL, [date\_generated] [datetime] NULL, CONSTRAINT [PK\_{0}\_ledger\_transactions\_staging] PRIMARY KEY CLUSTERED ( [id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]; ALTER TABLE {0}.[ledger\_transactions\_staging] WITH CHECK ADD CONSTRAINT

ALTER TABLE {0}.[ledger\_transactions\_staging] WITH CHECK ADD CONSTRAINT [FK\_{0}\_ledger\_transactions\_staging\_accounting\_periods] FOREIGN KEY([accounting\_period\_id]) REFERENCES {0}.[accounting\_periods] ([id]);

ALTER TABLE {0}.[ledger\_transactions\_staging] CHECK CONSTRAINT [FK\_{0}\_ledger\_transactions\_staging\_accounting\_periods];

END;

#### process\_logs

```
[process_id] [int] NOT NULL,
      [accounting_period_id] [int] NULL,
      [user_id] [int] NOT NULL,
      [date_executed] [datetime] NOT NULL,
      [descp] [varchar](100) NOT NULL,
      CONSTRAINT [PK_{0}_process_logs] PRIMARY KEY CLUSTERED
      (
        [id] ASC
      )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
      ) ON [PRIMARY];
```

END;

#### user\_bank\_columns

IF OBJECT\_ID(N'{0}.[user\_bank\_columns]', N'U') IS NULL BEGIN CREATE TABLE {0}.[user\_bank\_columns]( [user\_id] [int] NOT NULL, [branch] [bit] NOT NULL, [customer] [bit] NOT NULL, [account\_no] [bit] NOT NULL, [amount] [bit] NOT NULL, [currency] [bit] NOT NULL, [descp1] [bit] NOT NULL, [descp2] [bit] NOT NULL, [descp3] [bit] NOT NULL, [descp4] [bit] NOT NULL, [descp5] [bit] NOT NULL, [descp6] [bit] NOT NULL, [posting\_date] [bit] NOT NULL, [value\_date] [bit] NOT NULL, [clearing\_date] [bit] NOT NULL, [trans\_no] [bit] NOT NULL, [posting\_no] [bit] NOT NULL, [grouping] [bit] NOT NULL, [period] [bit] NOT NULL, [user\_comments] [bit] NOT NULL, CONSTRAINT [PK\_{0}\_user\_bank\_colunms] PRIMARY KEY CLUSTERED ( [user\_id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY];

END;

#### user\_ledger\_columns

IF OBJECT\_ID(N'{0}.[user\_ledger\_columns]', N'U') IS NULL BEGIN
 CREATE TABLE {0}.[user\_ledger\_columns](
 [user\_id] [int] NOT NULL,
 [jrnl\_id] [bit] NOT NULL,
 [ldr\_entity\_id] [bit] NOT NULL,
 [eff\_date] [bit] NOT NULL,

[jrnl\_seq\_nbr] [bit] NOT NULL, [ministry] [bit] NOT NULL, [program] [bit] NOT NULL, [subprog] [bit] NOT NULL, [account] [bit] NOT NULL, [project] [bit] NOT NULL, [sof] [bit] NOT NULL, [sector] [bit] NOT NULL, [posting\_yr] [bit] NOT NULL, [posting\_pd] [bit] NOT NULL, [dr\_cr\_code\_1] [bit] NOT NULL, [descp] [bit] NOT NULL, [trans\_amt] [bit] NOT NULL, [date\_posted] [bit] NOT NULL, [pmt\_meth\_id] [bit] NOT NULL, [pmt\_ref\_nbr] [bit] NOT NULL, [pmt\_ref\_date] [bit] NOT NULL, [bank\_id] [bit] NOT NULL, [bank\_acct\_nbr] [bit] NOT NULL, [vendor\_id] [bit] NOT NULL, [vendor\_loc\_code] [bit] NOT NULL, [vendor\_name] [bit] NOT NULL, [payable\_entity\_id] [bit] NOT NULL, [pmt\_rqst\_gross\_amt] [bit] NOT NULL, [pmt\_amt] [bit] NOT NULL, [type\_of\_pmt] [bit] NOT NULL, [grouping] [bit] NOT NULL, [period] [bit] NOT NULL, [user\_comments] [bit] NOT NULL, CONSTRAINT [PK\_{0}\_user\_ledger\_colunms] PRIMARY KEY CLUSTERED ( [user id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY];

END;

#### custom\_processes

IF OBJECT\_ID(N'{0}.[custom\_processes]', N'U') IS NULL BEGIN CREATE TABLE {0}.[custom\_processes]( [id] [int] IDENTITY(1,1) NOT NULL, [process] [varchar](50) NOT NULL, [process\_name] [varchar](50) NOT NULL, [hidden] bit NOT NULL, [parameter bit NOT NULL, [userid] bit NOT NULL, CONSTRAINT [PK\_{0}.custom\_processes] PRIMARY KEY CLUSTERED ( [id] ASC )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY]; END;", @"IF OBJECT\_ID(N'{0}.[custom\_reports]', N'U') IS NULL BEGIN CREATE TABLE {0}.[custom\_reports]( [id] [int] IDENTITY(1,1) NOT NULL, [report\_name] [varchar](50) NOT NULL, [display\_name] [varchar](50) NOT NULL,

```
[base_address] [varchar](100) NOT NULL,
      [url_setting] [varchar](100) NULL,
      [stored_procedure] [varchar](50) NULL,
      [parUser] [bit] NULL,
      [parDateRange] [bit] NULL,
      [parBankorLedger] [bit] NULL,
      [parBankGrouping] [bit] NULL,
      [parLedgerGrouping] [bit] NULL,
      [parAccountingPeriod] [bit] NULL,
      [parAccount] [bit] NULL,
CONSTRAINT [PK_{0}_custom_reports] PRIMARY KEY CLUSTERED
(
      [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY];
End;
```

#### reports\_history

```
IF OBJECT_ID(N'{0}.[reports_history]', N'U') IS NULL BEGIN
CREATE TABLE {0}.[reports_history](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [file_name] [varchar](50) NOT NULL,
    [accounting_period_id] [int] NOT NULL,
    [date_generated] [DateTime] NOT NULL,
    [generated_by] [int] NOT NULL,
    [generated_by] [int] NOT NULL,
    CONSTRAINT [PK_reports_history] PRIMARY KEY CLUSTERED
(
        [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
END;
```

# APPENDIX 3 – Scripts for Views

ledger\_closing\_balance\_vw

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE type = 'V' AND OBJECT\_ID = OBJECT\_ID('{0}.ledger\_closing\_balance\_vw')) BEGIN EXEC('Create view {0}.ledger\_closing\_balance\_vw as SELECT ldr\_entity\_id, processing\_yr, amt\_class\_type, ldr\_amt\_0, ldr\_amt\_1, ldr\_amt\_2, ldr\_amt\_3, ldr\_amt\_4, ldr\_amt\_5, ldr\_amt\_6, ldr\_amt\_7, ldr\_amt\_8, ldr\_amt\_9, ldr\_amt\_10, ldr\_amt\_11, ldr\_amt\_12, ldr\_amt\_13, ldr\_amt\_14 FROM {2}DBSglep.dbo.ldr\_acct\_bal AS lb amt\_class\_type = ''ACTUAL'' AND ldr\_entity\_id WHERE = ''D'' AND account = ''{1}'' and ministry=''31'' and program=''F500'' and subprog=''F51'';') END;

ledger\_trans\_sum\_vw

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE type = 'V' AND OBJECT\_ID =
OBJECT\_ID('{0}.ledger\_trans\_sum\_vw'))
BEGIN
EXEC('Create {0}.ledger\_trans\_sum\_vw as
select accounting\_period\_id, sum(trans\_amt) as amount
from {0}.ledger\_transactions
group by accounting\_period\_id') END;

bank\_trans\_sum\_vw

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE type = 'V' AND OBJECT\_ID =
OBJECT\_ID('{0}.ledger\_bank\_sum\_vw'))
BEGIN
EXEC('Create {0}.bank\_trans\_sum\_vw as
select accounting\_period\_id, sum(amount) as amount
from {0}.bank\_transactions
group by accounting\_period\_id') END;

# **APPENDIX 4 – Scripts for Stored Procedures**

bank\_staging\_process

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE type = 'P' AND OBJECT_ID =
OBJECT_ID('{0}.[bank_staging_process]'))
BEGIN
```

```
EXEC('
CREATE procedure {0}.[bank_staging_process]
                    (
                          @p_period int
                          ,@p_user int
                    )
                    as
                    declare @p_year int
                          ,@p_month int
                    begin
                          /* disable messages*/
                          set nocount on;
                          update {0}.[bank_transactions_staging]
                          set [bank_transaction_group_id]=1
                          where amount>0;
                          update {0}.[bank_transactions_staging]
                          set [bank_transaction_group_id]=2
                          where amount<0;
                    end;
')
End;
                           import_ledger_transactions
 IF NOT EXISTS (SELECT * FROM sys.objects WHERE type = 'P' AND OBJECT_ID =
OBJECT_ID('{0}.[import_ledger_transactions]'))
BEGIN
EXEC('CREATE procedure {0}.[import_ledger_transactions]
(
      @p_period int
      ,@p_user int
)
as
declare @p_year int
    ,@p_month int
begin
      /* disable messages*/
    set nocount on;
      /*Get the year and month value*/
      select @p_month=posting_pd, @p_year=posting_year
      from {0}.accounting_periods
      where id=@p_period;
      /* Import the transactions*/
 select
jl.jrnl_id,ldr_entity_id,eff_date,jl.jrnl_seq_nbr,ministry,program,subprog,account,p
roject, sof, sector,
                    posting_yr,posting_pd,dr_cr_code_1,descp,trans_amt,date_posted,
      pmt.pmt_meth_id,pmt.pmt_ref_nbr,pmt.pmt_ref_date,pmt.bank_id,pmt.bank_acct_nb
\mathbf{r}
                                 ,pmt.vendor_id
,pmt.vendor_loc_code,pmt.vendor_name,pmt.payable_entity_id
,pmt.pmt_rqst_gross_amt,pmt.pmt_amt,pmt.type_of_pmt,
```

```
@p_period as accounting_period_id,0 as matched,0 as
is_locked,GETDATE() as date_imported,@p_user as import_user
                                 into #tmp_ledger_transaction_prestaging
                                 from {2}DBSjepc.dbo.posted_jrnl_line jl
                                 Left Join {2}DBSpymt.dbo.pmt pmt on pmt.jrnl_id =
jl.jrnl_id
                                 where posting_yr=@p_year
                                 and posting_pd=@p_month
                                 and ldr_entity_id =''D''
                                 and ministry=''31''
                                 and program=''F500''
                                 and subprog=''F51''
                                 and account=''{1}'';
             /*Remove duplicates and insert into staging*/
      insert into {0}.ledger_transactions_staging([jrnl_id]
           ,[ldr_entity_id]
           ,[eff_date]
           ,[jrnl_seq_nbr]
           ,[ministry]
           ,[program]
           ,[subprog]
           ,[account]
           ,[project]
           ,[sof]
           ,[sector]
           ,[posting_yr]
           ,[posting_pd]
           ,[dr_cr_code_1]
           ,[descp]
           ,[trans_amt]
           ,[date_posted]
           ,[pmt_meth_id]
           ,[pmt_ref_nbr]
           ,[pmt_ref_date]
           ,[bank_id]
           ,[bank_acct_nbr]
           ,[vendor_id]
           ,[vendor_loc_code]
           ,[vendor_name]
           ,[payable_entity_id]
           ,[pmt_rqst_gross_amt]
           ,[pmt_amt]
           ,[type_of_pmt]
           ,[accounting_period_id]
           ,[matched]
           ,[is_locked]
           ,[date_imported]
           ,[import_user])
 select [jrnl_id]
           ,[ldr_entity_id]
           ,[eff_date]
           ,[jrnl_seq_nbr]
           ,[ministry]
           ,[program]
```

```
,[subprog]
           ,[account]
           ,[project]
           ,[sof]
           ,[sector]
           ,[posting_yr]
           ,[posting_pd]
           ,[dr_cr_code_1]
           ,[descp]
           ,[trans_amt]
           ,[date_posted]
           ,[pmt_meth_id]
           ,[pmt_ref_nbr]
           ,[pmt_ref_date]
           ,[bank_id]
           ,[bank_acct_nbr]
           ,[vendor_id]
           ,[vendor_loc_code]
           ,[vendor_name]
           ,[payable_entity_id]
           ,[pmt_rqst_gross_amt]
           ,[pmt_amt]
           ,[type_of_pmt]
           ,[accounting_period_id]
           ,[matched]
           ,[is_locked]
           ,[date_imported]
           ,[import_user]
                from #tmp_ledger_transaction_prestaging
                where jrnl_id COLLATE SQL_Latin1_General_CP1_CI_AS not in (select
jrnl_id from {0}.ledger_transactions_staging)
                and jrnl_id COLLATE SQL_Latin1_General_CP1_CI_AS not in (select
jrnl_id from {0}.ledger_transactions);
      /* drop the temp table*/
      drop table #tmp_ledger_transaction_prestaging;
end;')
End;
                             ledger_staging_process
            IF NOT EXISTS (SELECT * FROM sys.objects WHERE type = 'P' AND OBJECT_ID
= OBJECT_ID('{0}.[ledger_staging_process]'))
BEGIN
EXEC('CREATE procedure {0}.[ledger_staging_process]
```

```
set ledger_transaction_group_id=1
where trans_amt>0;
update {0}.ledger_transactions_staging
set ledger_transaction_group_id=2
where trans_amt<0;
end;')
End;</pre>
```

#### generate\_ss\_flLe

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE type = 'P' AND OBJECT_ID
= OBJECT_ID('{0}.[generate_ss_fILe]'))
BEGIN
EXEC('Create procedure {0}.[generate_ss_fILe] as
begin
select [type_of_pmt] as pmt_type
       ,convert(varchar, bt.posting_date, 110) tdate,
         CAST(lp.pmt_amt as varchar) amt,
         CAST(lp.pmt_ref_nbr as varchar) ref_nbr
         ,lp.id as id
         into #temp_clearance
        from {0}.ledger_transactions lp, {0}.bank_transactions bt
        where lp.matched=1
        and bt.descp1=cast(lp.pmt_ref_nbr as varchar)
        and bt.bank_transaction_group_id in (0)--edit the group
        and lp.pmt_amt=(bt.amount)
        and (lp.generated=0 or lp.generated is null);
update {0}.ledger_transactions
set generated=1,date_generated=GETDATE()
where id in (select id from #temp_clearance);
--edit the bank info
select ''/*bank,bank_no,{1},PC,''+
ISNULL(pmt_type,'''')+'',''+ISNULL(tdate,'''')+'',''+ISNULL(amt,'''')+'',''+ISNULL(r
ef_nbr,''')
```

```
from #temp_clearance;
End;')
End;
```

#### matching\_process

```
DECLARE @LedgerId int
DECLARE @BankGroup varchar(50)
DECLARE bank_cursor CURSOR FOR
SELECT bt.id, bt.amount, bt.descp1, LOWER(g.descp)
FROM {0}.[bank_transactions] bt
LEFT JOIN {0}.[bank_transaction_groups] g on bt.bank_transaction_group_id=g.id
WHERE matched is null or matched=0
OPEN bank_cursor
FETCH NEXT FROM bank_cursor INTO @BankId,@Amount,@ValToCompare,@BankGroup
WHILE @@FETCH_STATUS = 0
BEGIN
      /*Make sure we only match to 1 record, let us do a check*/
      DECLARE ledger_cursor CURSOR STATIC FOR
      SELECT lt.id
      FROM {0}.[ledger_transactions] lt
      LEFT JOIN {0}.[ledger_transaction_groups] g on
lt.ledger_transaction_group_id=g.id
      WHERE trans_amt=@Amount
      and LOWER(g.descp)=@BankGroup
      /*<----IF WE GOT RULES FOR COMPARISM HERE*/
      OPEN ledger_cursor
   /*if we got 1 row then we match, if not then move to the next transaction*/
      IF @@Cursor_Rows=1
      BEGIN
        FETCH NEXT FROM ledger_cursor INTO @LedgerId
        /*Maks surs we update both*/
        BEGIN TRANSACTION
        /*Update bank*/
        UPDATE {0}.[bank_transactions]
        SET matched=1
        ,is_locked=0
        ,lock_user=0
        ,date_matched=GETDATE()
        ,matched_by=-1
        ,[matched_bank_transactions]=0
        ,[matched_ledger_transactions]=@LedgerId
        WHERE id =@BankId
        /*Update the ledger*/
        UPDATE {0}.[ledger_transactions]
        SET matched=1
        ,is_locked=0
        ,lock_user=0
        ,date_matched=GETDATE()
        ,matched_by=-1
        ,[matched_bank_transactions]=@BankId
        ,[matched_ledger_transactions]=0
        WHERE id =@LedgerId
        COMMIT;
```

END CLOSE ledger\_cursor DEALLOCATE ledger\_cursor FETCH NEXT FROM bank\_cursor INTO @BankId,@Amount,@ValToCompare,@BankGroup

END

CLOSE bank\_cursor DEALLOCATE bank\_cursor /\*FIN\*/ end;') End;