# AccuFund to Nonprofit Fund Accounting System v8.8 - Verification Procedure

# AccuFund to Nonprofit Fund Accounting System (v8.8) Verification Procedure

**Document Version:** 1.0

**Last Updated:** July 11, 2025

**Applicable System Versions:** AccuFund (All Versions), Nonprofit Fund Accounting System (v8.8)

## Table of Contents

# 1. Introduction and Methodology

## 1.1 Purpose

This document provides a comprehensive, step-by-step procedure for verifying that the Nonprofit Fund Accounting System (v8.8) produces results identical to AccuFund. The verification process is crucial during migration to ensure data integrity, financial accuracy, and continuity of financial reporting.

## 1.2 Verification Methodology

The verification methodology follows a systematic approach:

1. **Structural Verification:** Ensuring the chart of accounts, funds, entities, and other structural elements are correctly migrated.
2. **Transactional Verification:** Confirming that individual transactions are accurately represented in the new system.
3. **Balance Verification:** Validating that account balances, fund balances, and financial statements match between systems.
4. **Special Feature Verification:** Specific testing for v8.8 features like Inter-Entity Transfers and Bank Account Management.
5. **Report Verification:** Comparing standard and custom reports between systems.

## 1.3 Verification Scope

This verification procedure covers:

- Chart of Accounts, Funds, and Entities
- General Ledger Transactions
- Accounts Payable and Accounts Receivable
- Fund Balances and Restrictions
- Inter-Entity Transfers
- Bank Account Management
- Grant Tracking
- Financial Statements and Reports

## 1.4 Required Resources

To complete this verification, you will need:

- Access to both AccuFund and Nonprofit Fund Accounting System (v8.8)
- Database query access to both systems (for SQL queries)
- Microsoft Excel or similar spreadsheet software

- Exported reports from both systems
- Verification templates (provided in this document)
- Staff familiar with both systems

# 2. Pre-Verification Preparation

## 2.1 Establish a Verification Point

1Select a Verification Date

Choose a specific date (typically month-end or year-end) as your verification point. All comparisons will be made as of this date.

**Recommendation:** Select the most recent month-end where AccuFund has been fully reconciled and closed.

2Freeze AccuFund Activity

Ensure no further transactions are entered in AccuFund for periods being verified. If ongoing operations require continued use of AccuFund, document a clear cutoff point.

3Create Verification Database Backups

Create backups of both systems' databases at the verification point:

```
-- AccuFund Backup (if using SQL Server)
BACKUP DATABASE AccuFundDB TO DISK = 'C:\Backups\AccuFundDB_Verification.bak';

-- Nonprofit Fund Accounting System Backup (PostgreSQL)
pg_dump -U postgres fund_accounting_db > /backups/fund_accounting_verification.sql
```

## 2.2 Prepare Verification Templates

1Create Master Verification Checklist

Create a master spreadsheet with tabs for each verification area:

- Chart of Accounts Verification
- Fund Verification
- Entity Verification
- Transaction Verification
- Balance Verification
- Inter-Entity Transfer Verification

- Bank Account Verification
- Grant Verification
- Report Comparison

2Create Variance Tracking Template

Create a template for tracking and investigating variances:

| Item ID | Description | AccuFund Value | New System Value | Variance | Investigation Notes | Resolution | Status |
|---|---|---|---|---|---|---|---|
| VAR-001 | [Example] Cash Account Balance | $10,000.00 | $9,950.00 | $50.00 | Bank fee recorded in AccuFund but not migrated | Added missing transaction | Resolved |

## 2.3 Export Reference Data

1Export AccuFund Reference Data

Export the following data from AccuFund:

- Chart of Accounts listing
- Fund listing
- Entity/Department listing
- Bank Account listing
- Trial Balance as of verification date
- General Ledger detail for the verification period
- Standard financial reports (Balance Sheet, Income Statement, etc.)

2Export Nonprofit Fund Accounting System Data

Export the same data sets from the new system for comparison.

# 3. Account Structure Verification

## 3.1 Chart of Accounts Verification

1Compare Account Counts

Verify the total number of accounts matches between systems:

-- AccuFund Query (SQL Server example)

```
SELECT COUNT(*) AS TotalAccounts FROM ChartOfAccounts WHERE Status = 'Active';

-- Nonprofit Fund Accounting System Query (PostgreSQL)
SELECT COUNT(*) AS total_accounts FROM accounts WHERE status = 'active';
```

2Create Account Comparison Spreadsheet

Export both charts of accounts to Excel and use VLOOKUP or INDEX/MATCH to compare:

```
=IF(ISNA(VLOOKUP(A2,AccuFundAccounts!$A$2:$B$1000,2,FALSE)),"Missing in AccuFund",
IF(VLOOKUP(A2,AccuFundAccounts!$A$2:$B$1000,2,FALSE)=B2,"Match","Different Name"))
```

Include the following fields in your comparison:

- Account Code/Number
- Account Name
- Account Type (Asset, Liability, etc.)
- Account Subtype
- Status (Active/Inactive)
- Parent Account (if hierarchical)

3Verify Account Hierarchies

Ensure account hierarchies and parent-child relationships are preserved:

```
-- AccuFund Query
SELECT ChildAccount, ParentAccount FROM AccountHierarchy;

-- Nonprofit Fund Accounting System Query
SELECT a.code AS child_account, p.code AS parent_account
FROM accounts a
LEFT JOIN accounts p ON a.parent_id = p.id
WHERE a.parent_id IS NOT NULL;
```

## 3.2 Fund Structure Verification

1Compare Fund Counts

Verify the total number of funds matches between systems:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalFunds FROM Funds WHERE Status = 'Active';

-- Nonprofit Fund Accounting System Query
```

```
SELECT COUNT(*) AS total_funds FROM funds WHERE status = 'active';
```

2Create Fund Comparison Spreadsheet

Export both fund lists to Excel and compare:

| Fund Code | AccuFund Name | New System Name | AccuFund Type | New System Type | Status |
|---|---|---|---|---|---|
| 101 | General Operating | General Operating | Unrestricted | Unrestricted | Match |

3Verify Fund Restrictions

Ensure fund restrictions are correctly migrated:

```
-- AccuFund Query
SELECT FundCode, RestrictionType FROM Funds;

-- Nonprofit Fund Accounting System Query
SELECT code, restriction_type FROM funds;
```

Verify that restriction types map correctly between systems:

- Unrestricted
- Temporarily Restricted (or With Donor Restrictions - Time/Purpose)
- Permanently Restricted (or With Donor Restrictions - Perpetual)

## 3.3 Entity Structure Verification

1Compare Entity Counts

Verify the total number of entities/departments matches between systems:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalEntities FROM Departments WHERE Status = 'Active';

-- Nonprofit Fund Accounting System Query
SELECT COUNT(*) AS total_entities FROM entities WHERE status = 'active';
```

2Create Entity Comparison Spreadsheet

Export both entity lists to Excel and compare all attributes.

3Verify Entity Hierarchies

Ensure entity hierarchies and parent-child relationships are preserved:

```
-- AccuFund Query
SELECT ChildDept, ParentDept FROM DepartmentHierarchy;
```

```
-- Nonprofit Fund Accounting System Query
SELECT e.code AS child_entity, p.code AS parent_entity
FROM entities e
LEFT JOIN entities p ON e.parent_id = p.id
WHERE e.parent_id IS NOT NULL;
```

# 4. Transaction Reconciliation Process

## 4.1 General Ledger Transaction Verification

1Compare Transaction Counts

Verify the total number of transactions for the verification period:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalTransactions
FROM JournalEntries
WHERE EntryDate BETWEEN '2025-01-01' AND '2025-06-30';

-- Nonprofit Fund Accounting System Query
SELECT COUNT(*) AS total_transactions
FROM journal_entries
WHERE entry_date BETWEEN '2025-01-01' AND '2025-06-30';
```

2Compare Transaction Totals

Verify the total debits and credits match for the verification period:

```
-- AccuFund Query
SELECT SUM(DebitAmount) AS TotalDebits, SUM(CreditAmount) AS TotalCredits
FROM JournalEntryLines
WHERE EntryDate BETWEEN '2025-01-01' AND '2025-06-30';

-- Nonprofit Fund Accounting System Query
SELECT SUM(jel.debit_amount) AS total_debits, SUM(jel.credit_amount) AS total_credits
FROM journal_entry_lines jel
JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE je.entry_date BETWEEN '2025-01-01' AND '2025-06-30';
```

3Sample Transaction Verification

Select a representative sample of transactions for detailed verification:

- High-value transactions

- Transactions with multiple line items
- Transactions involving multiple funds
- Transactions involving multiple entities
- Transactions with unusual account combinations

For each sampled transaction, verify:

- Transaction date
- Reference number/description
- Debit and credit amounts
- Account assignments
- Fund assignments
- Entity assignments
- Supporting documentation references

## 4.2 Accounts Payable Verification

1Compare Vendor Counts and Details

Verify vendor information has been correctly migrated:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalVendors FROM Vendors WHERE Status = 'Active';

-- Nonprofit Fund Accounting System Query
SELECT COUNT(*) AS total_vendors FROM vendors WHERE status = 'active';
```

Sample key vendors and verify all details match between systems.

2Compare Open AP Invoices

Verify all open invoices match between systems:

```
-- AccuFund Query
SELECT VendorID, InvoiceNumber, InvoiceDate, DueDate, InvoiceAmount, RemainingAmount
FROM APInvoices
WHERE Status = 'Open';

-- Nonprofit Fund Accounting System Query
SELECT vendor_id, invoice_number, invoice_date, due_date, invoice_amount, remaining_amount
FROM ap_invoices
WHERE status = 'open';
```

3Verify AP Aging

Compare AP aging reports between systems to ensure they match.

### 4.3 Accounts Receivable Verification

1Compare Customer Counts and Details

Verify customer information has been correctly migrated.

2Compare Open AR Invoices

Verify all open receivables match between systems.

3Verify AR Aging

Compare AR aging reports between systems to ensure they match.

# 5. Balance Verification

## 5.1 Trial Balance Verification

1Generate Trial Balances

Generate trial balances from both systems as of the verification date:

```
-- AccuFund Query
SELECT a.AccountCode, a.AccountName,
SUM(CASE WHEN jel.DebitAmount > 0 THEN jel.DebitAmount ELSE 0 END) AS Debits,
SUM(CASE WHEN jel.CreditAmount > 0 THEN jel.CreditAmount ELSE 0 END) AS Credits,
SUM(jel.DebitAmount - jel.CreditAmount) AS Balance
FROM ChartOfAccounts a
LEFT JOIN JournalEntryLines jel ON a.AccountID = jel.AccountID
WHERE jel.EntryDate <= '2025-06-30'
GROUP BY a.AccountCode, a.AccountName
ORDER BY a.AccountCode;

-- Nonprofit Fund Accounting System Query
SELECT a.code AS account_code, a.name AS account_name,
SUM(CASE WHEN jel.debit_amount > 0 THEN jel.debit_amount ELSE 0 END) AS debits,
SUM(CASE WHEN jel.credit_amount > 0 THEN jel.credit_amount ELSE 0 END) AS credits,
SUM(jel.debit_amount - jel.credit_amount) AS balance
FROM accounts a
```

```
LEFT JOIN journal_entry_lines jel ON a.id = jel.account_id
LEFT JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE je.entry_date <= '2025-06-30'
GROUP BY a.code, a.name
ORDER BY a.code;
```

2Create Trial Balance Comparison Spreadsheet

Create a spreadsheet that compares account balances side by side:

| Account Code | Account Name | AccuFund Balance | New System Balance | Variance | Variance % | Status |
|---|---|---|---|---|---|---|
| 1000 | Cash - Operating | $125,432.78 | $125,432.78 | $0.00 | 0.00% | Match |

Add a formula to calculate variance and variance percentage:

Variance: =C2-D2
Variance %: =IF(C2=0,IF(D2=0,"0.00%","-"),E2/C2)
3Verify Trial Balance Totals

Confirm that the trial balance totals match between systems:

- Total Debits should match
- Total Credits should match
- Net difference should be zero in both systems

## 5.2 Balance Sheet Verification

1Generate Balance Sheets

Generate balance sheets from both systems as of the verification date.

2Compare Key Balance Sheet Categories

Create a comparison of key balance sheet categories:

| Category | AccuFund Balance | New System Balance | Variance | Status |
|---|---|---|---|---|
| Total Current Assets | $450,000.00 | $450,000.00 | $0.00 | Match |
| Total Fixed Assets | $750,000.00 | $750,000.00 | $0.00 | Match |
| Total Assets | $1,200,000.00 | $1,200,000.00 | $0.00 | Match |
| Total Current Liabilities | $175,000.00 | $175,000.00 | $0.00 | Match |
| Total Long-term Liabilities | $300,000.00 | $300,000.00 | $0.00 | Match |

| Category | AccuFund Balance | New System Balance | Variance | Status |
|---|---|---|---|---|
| Total Liabilities | $475,000.00 | $475,000.00 | $0.00 | Match |
| Net Assets Without Donor Restrictions | $500,000.00 | $500,000.00 | $0.00 | Match |
| Net Assets With Donor Restrictions | $225,000.00 | $225,000.00 | $0.00 | Match |
| Total Net Assets | $725,000.00 | $725,000.00 | $0.00 | Match |
| Total Liabilities and Net Assets | $1,200,000.00 | $1,200,000.00 | $0.00 | Match |

## 5.3 Income Statement Verification

1Generate Income Statements

Generate income statements (Statement of Activities) from both systems for the verification period.

2Compare Key Income Statement Categories

Create a comparison of key income statement categories:

| Category | AccuFund Amount | New System Amount | Variance | Status |
|---|---|---|---|---|
| Total Revenue Without Donor Restrictions | $850,000.00 | $850,000.00 | $0.00 | Match |
| Total Revenue With Donor Restrictions | $250,000.00 | $250,000.00 | $0.00 | Match |
| Total Revenue | $1,100,000.00 | $1,100,000.00 | $0.00 | Match |
| Program Expenses | $600,000.00 | $600,000.00 | $0.00 | Match |
| Administrative Expenses | $150,000.00 | $150,000.00 | $0.00 | Match |
| Fundraising Expenses | $100,000.00 | $100,000.00 | $0.00 | Match |
| Total Expenses | $850,000.00 | $850,000.00 | $0.00 | Match |
| Change in Net Assets | $250,000.00 | $250,000.00 | $0.00 | Match |

## 5.4 Fund Balance Verification

1Generate Fund Balance Reports

Generate fund balance reports from both systems as of the verification date:

```
-- AccuFund Query
SELECT f.FundCode, f.FundName,
SUM(jel.DebitAmount - jel.CreditAmount) AS FundBalance
FROM Funds f
LEFT JOIN JournalEntryLines jel ON f.FundID = jel.FundID
WHERE jel.EntryDate <= '2025-06-30'
GROUP BY f.FundCode, f.FundName
ORDER BY f.FundCode;

-- Nonprofit Fund Accounting System Query
SELECT f.code AS fund_code, f.name AS fund_name,
SUM(jel.debit_amount - jel.credit_amount) AS fund_balance
FROM funds f
LEFT JOIN journal_entry_lines jel ON f.id = jel.fund_id
LEFT JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE je.entry_date <= '2025-06-30'
GROUP BY f.code, f.name
ORDER BY f.code;
```

2Compare Fund Balances

Create a spreadsheet comparing fund balances between systems.

# 6. Inter-Entity Transfer Verification (v8.8)

## 6.1 Inter-Entity Transfer Structure Verification

1Verify Due To/Due From Account Setup

Ensure that Due To/Due From accounts are correctly configured in the new system:

```
-- Nonprofit Fund Accounting System Query
SELECT e1.code AS entity_code, e1.name AS entity_name,
a_due_from.code AS due_from_account_code, a_due_from.name AS due_from_account_name,
a_due_to.code AS due_to_account_code, a_due_to.name AS due_to_account_name
FROM entities e1
JOIN entities e2 ON e1.id != e2.id
LEFT JOIN accounts a_due_from ON a_due_from.entity_id = e1.id AND
a_due_from.name LIKE 'Due From ' || e2.name
LEFT JOIN accounts a_due_to ON a_due_to.entity_id = e1.id AND a_due_to.name
LIKE 'Due To ' || e2.name
ORDER BY e1.code, e2.code;
```

Verify that:

- Each entity has a "Due From" account (asset, typically 19xx series) for each counterparty entity
- Each entity has a "Due To" account (liability, typically 29xx series) for each counterparty entity
- The naming conventions are consistent

## 6.2 Inter-Entity Transfer Transaction Verification

1Identify Inter-Entity Transfers in AccuFund

Identify transactions in AccuFund that represent inter-entity transfers:

```
-- AccuFund Query (example - may need customization based on AccuFund setup)
SELECT je.JournalEntryID, je.EntryDate, je.Reference, jel.DebitAmount, jel.CreditAmount,
a.AccountCode, a.AccountName, e.EntityCode, e.EntityName
FROM JournalEntries je
JOIN JournalEntryLines jel ON je.JournalEntryID = jel.JournalEntryID
JOIN ChartOfAccounts a ON jel.AccountID = a.AccountID
JOIN Entities e ON jel.EntityID = e.EntityID
WHERE (a.AccountCode LIKE '19%' OR a.AccountCode LIKE '29%')
AND (a.AccountName LIKE 'Due To%' OR a.AccountName LIKE 'Due From%')
AND je.EntryDate BETWEEN '2025-01-01' AND '2025-06-30'
ORDER BY je.EntryDate, je.JournalEntryID;
```

2Identify Inter-Entity Transfers in New System

Identify inter-entity transfers in the Nonprofit Fund Accounting System:

```
-- Nonprofit Fund Accounting System Query
SELECT je.id, je.entry_date, je.reference_number, je.is_inter_entity,
je.matching_transaction_id, je.target_entity_id,
jel.debit_amount, jel.credit_amount,
a.code AS account_code, a.name AS account_name,
e.code AS entity_code, e.name AS entity_name
FROM journal_entries je
JOIN journal_entry_lines jel ON je.id = jel.journal_entry_id
JOIN accounts a ON jel.account_id = a.id
JOIN entities e ON je.entity_id = e.id
WHERE je.is_inter_entity = true
AND je.entry_date BETWEEN '2025-01-01' AND '2025-06-30'
ORDER BY je.entry_date, je.id;
```

3Compare Inter-Entity Transfer Counts and Amounts

Verify that the total number and amounts of inter-entity transfers match between systems.

4Verify Paired Transactions

Ensure that each inter-entity transfer in the new system consists of properly paired transactions:

```
-- Nonprofit Fund Accounting System Query
SELECT je1.id AS transaction1_id, je1.reference_number AS transaction1_ref,
e1.code AS source_entity, je1.entry_date,
je2.id AS transaction2_id, je2.reference_number AS transaction2_ref,
e2.code AS target_entity,
je1.matching_transaction_id
FROM journal_entries je1
JOIN journal_entries je2 ON je1.matching_transaction_id = je2.matching_transaction_id
AND je1.id != je2.id
JOIN entities e1 ON je1.entity_id = e1.id
JOIN entities e2 ON je2.entity_id = e2.id
WHERE je1.is_inter_entity = true
AND je1.entry_date BETWEEN '2025-01-01' AND '2025-06-30'
ORDER BY je1.entry_date, je1.id;
```

Verify that:

- Each inter-entity transaction has a matching pair
- The matching_transaction_id links the pairs correctly
- The amounts balance between the pairs
- The correct Due To/Due From accounts are used

## 6.3 Inter-Entity Balance Verification

1Verify Inter-Entity Account Balances

Ensure that Due To/Due From account balances match between entities:

```
-- Nonprofit Fund Accounting System Query
SELECT e1.code AS entity1_code, e1.name AS entity1_name,
e2.code AS entity2_code, e2.name AS entity2_name,
a_due_from.code AS due_from_account, SUM(jel_from.debit_amount - jel_from.credit_amount) AS due_from_balance,
a_due_to.code AS due_to_account, SUM(jel_to.credit_amount - jel_to.debit_amount)
AS due_to_balance
FROM entities e1
JOIN entities e2 ON e1.id != e2.id
JOIN accounts a_due_from ON a_due_from.entity_id = e1.id AND a_due_from.name
LIKE 'Due From ' || e2.name
JOIN accounts a_due_to ON a_due_to.entity_id = e2.id AND a_due_to.name LIKE 'Due
To ' || e1.name
LEFT JOIN journal_entry_lines jel_from ON jel_from.account_id = a_due_from.id
LEFT JOIN journal_entry_lines jel_to ON jel_to.account_id = a_due_to.id
LEFT JOIN journal_entries je_from ON jel_from.journal_entry_id = je_from.id
LEFT JOIN journal_entries je_to ON jel_to.journal_entry_id = je_to.id
WHERE je_from.entry_date <= '2025-06-30' AND je_to.entry_date <= '2025-06-30'
GROUP BY e1.code, e1.name, e2.code, e2.name, a_due_from.code, a_due_to.code
ORDER BY e1.code, e2.code;
```

Verify that for each entity pair:

- Entity A's "Due From Entity B" balance equals Entity B's "Due To Entity A" balance
- The balances match what was in AccuFund

2Test Inter-Entity Transfer Wizard

Create a test inter-entity transfer using the new wizard and verify:

- Both sides of the transaction are created correctly
- Due To/Due From accounts are used properly
- The transactions are linked with a matching_transaction_id
- The is_inter_entity flag is set to true
- The target_entity_id is set correctly

# 7. Bank Account Verification (v8.7+)

## 7.1 Bank Account Structure Verification

1Compare Bank Account Counts

Verify the total number of bank accounts matches between systems:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalBankAccounts FROM BankAccounts WHERE Status =
'Active';

-- Nonprofit Fund Accounting System Query
SELECT COUNT(*) AS total_bank_accounts FROM bank_accounts WHERE status =
'Active';
```

2Create Bank Account Comparison Spreadsheet

Export bank account details from both systems and compare:

| Bank Name | Account Name | AccuFund Account Number | New System Account Number | Account Type | Status | Match Status |
|---|---|---|---|---|---|---|
| First Na- | Operating | 123456789 | 123456789 | Checking | Active | Match |

| Bank Name | Account Name | AccuFund Account Number | New System Account Number | Account Type | Status | Match Status |
|---|---|---|---|---|---|---|
| tional Bank | Checking | | | | | |

3Verify Bank Account to GL Account Mapping

Ensure that bank accounts are correctly mapped to corresponding GL accounts:

```
-- Nonprofit Fund Accounting System Query
SELECT ba.bank_name, ba.account_name, ba.account_number,
a.code AS gl_account_code, a.name AS gl_account_name
FROM bank_accounts ba
JOIN accounts a ON ba.gl_account_id = a.id
ORDER BY ba.bank_name, ba.account_name;
```

Verify that:

- Each bank account is mapped to the correct GL account
- The GL account type is appropriate for the bank account type (e.g., checking accounts mapped to cash accounts)
- The mapping matches what was in AccuFund

## 7.2 Bank Account Balance Verification

1Compare Bank Account Balances

Verify that bank account balances match between systems:

```
-- AccuFund Query
SELECT ba.BankName, ba.AccountName, ba.AccountNumber, ba.Balance
FROM BankAccounts ba
WHERE ba.Status = 'Active'
ORDER BY ba.BankName, ba.AccountName;

-- Nonprofit Fund Accounting System Query
SELECT bank_name, account_name, account_number, balance
FROM bank_accounts
WHERE status = 'Active'
ORDER BY bank_name, account_name;
```

2Verify Bank Balance to GL Balance Reconciliation

Ensure that bank account balances reconcile with corresponding GL account balances:

```
-- Nonprofit Fund Accounting System Query
```

```
SELECT ba.bank_name, ba.account_name, ba.balance AS bank_balance,
SUM(jel.debit_amount - jel.credit_amount) AS gl_balance,
ba.balance - SUM(jel.debit_amount - jel.credit_amount) AS difference
FROM bank_accounts ba
JOIN accounts a ON ba.gl_account_id = a.id
LEFT JOIN journal_entry_lines jel ON jel.account_id = a.id
LEFT JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE je.entry_date <= '2025-06-30'
GROUP BY ba.bank_name, ba.account_name, ba.balance
ORDER BY ba.bank_name, ba.account_name;
```

Verify that:

- Bank account balances match corresponding GL account balances
- Any differences are explained by reconciling items (e.g., outstanding checks, deposits in transit)

## 7.3 Bank Account Reconciliation Verification

1Verify Last Reconciliation Date

Ensure that the last reconciliation date for each bank account matches between systems:

```
-- AccuFund Query
SELECT ba.BankName, ba.AccountName, ba.LastReconciliationDate
FROM BankAccounts ba
WHERE ba.Status = 'Active'
ORDER BY ba.BankName, ba.AccountName;

-- Nonprofit Fund Accounting System Query
SELECT bank_name, account_name, last_reconciliation_date
FROM bank_accounts
WHERE status = 'Active'
ORDER BY bank_name, account_name;
```

2Verify Reconciliation History

Compare reconciliation history between systems:

```
-- Nonprofit Fund Accounting System Query
SELECT ba.bank_name, ba.account_name,
br.reconciliation_date, br.statement_balance, br.cleared_balance,
br.uncleared_deposits, br.uncleared_withdrawals
FROM bank_reconciliations br
JOIN bank_accounts ba ON br.bank_account_id = ba.id
ORDER BY ba.bank_name, ba.account_name, br.reconciliation_date DESC;
```

3Test Bank Reconciliation Process

Perform a test bank reconciliation in the new system and verify:

- The reconciliation process works correctly
- Cleared and uncleared items are tracked properly
- The reconciliation report matches what would be produced in AccuFund
- The reconciliation history is updated correctly

# 8. Grant and Fund Restriction Verification

## 8.1 Grant Structure Verification

1Compare Grant Counts and Details

Verify grant information has been correctly migrated:

```
-- AccuFund Query
SELECT COUNT(*) AS TotalGrants FROM Grants WHERE Status = 'Active';

-- Nonprofit Fund Accounting System Query
SELECT COUNT(*) AS total_grants FROM grants WHERE status = 'active';
```

2Create Grant Comparison Spreadsheet

Export grant details from both systems and compare:

| Grant ID | AccuFund Name | New System Name | Start Date | End Date | Award Amount | Status |
|---|---|---|---|---|---|---|

## 8.2 Fund Restriction Verification

1Verify Fund Restriction Classifications

Ensure fund restrictions are correctly classified in the new system:

```
-- Nonprofit Fund Accounting System Query
SELECT restriction_type, COUNT(*) AS fund_count
FROM funds
GROUP BY restriction_type;
```

2Verify Restricted Fund Balances

Compare restricted fund balances between systems:

```
-- AccuFund Query
SELECT 'Without Donor Restrictions' AS RestrictionType,
SUM(jel.DebitAmount - jel.CreditAmount) AS TotalBalance
FROM JournalEntryLines jel
```

```
JOIN Funds f ON jel.FundID = f.FundID
WHERE f.RestrictionType = 'Unrestricted'
AND jel.EntryDate <= '2025-06-30'
UNION
SELECT 'With Donor Restrictions' AS RestrictionType,
SUM(jel.DebitAmount - jel.CreditAmount) AS TotalBalance
FROM JournalEntryLines jel
JOIN Funds f ON jel.FundID = f.FundID
WHERE f.RestrictionType IN ('Temporarily Restricted', 'Permanently Restricted')
AND jel.EntryDate <= '2025-06-30';

-- Nonprofit Fund Accounting System Query
SELECT
CASE
WHEN f.restriction_type = 'unrestricted' THEN 'Without Donor Restrictions'
ELSE 'With Donor Restrictions'
END AS restriction_type,
SUM(jel.debit_amount - jel.credit_amount) AS total_balance
FROM journal_entry_lines jel
JOIN funds f ON jel.fund_id = f.id
JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE je.entry_date <= '2025-06-30'
GROUP BY
CASE
WHEN f.restriction_type = 'unrestricted' THEN 'Without Donor Restrictions'
ELSE 'With Donor Restrictions'
END;
```

## 8.3 Grant Budget Verification

1Compare Grant Budgets

Verify grant budgets match between systems:

```
-- AccuFund Query
SELECT g.GrantID, g.GrantName, gb.AccountCode, gb.BudgetAmount
FROM Grants g
JOIN GrantBudgets gb ON g.GrantID = gb.GrantID
ORDER BY g.GrantID, gb.AccountCode;

-- Nonprofit Fund Accounting System Query
SELECT g.id AS grant_id, g.name AS grant_name, a.code AS account_code, gb.budget_amount
FROM grants g
JOIN grant_budgets gb ON g.id = gb.grant_id
JOIN accounts a ON gb.account_id = a.id
```

```
ORDER BY g.id, a.code;
```

2Verify Grant Actuals vs. Budget

Compare grant budget vs. actual reports between systems.

# 9. Net Assets and Fund Balances Reconciliation

## 9.1 Net Asset Classification Verification

1Verify Net Asset Classifications

Ensure net asset classifications match between systems:

```
-- AccuFund Query
SELECT
CASE
WHEN f.RestrictionType = 'Unrestricted' THEN 'Without Donor Restrictions'
WHEN f.RestrictionType = 'Temporarily Restricted' THEN 'With Donor Restrictions - Time/Purpose'
WHEN f.RestrictionType = 'Permanently Restricted' THEN 'With Donor Restrictions - Perpetual'
END AS NetAssetClass,
SUM(jel.DebitAmount - jel.CreditAmount) AS Balance
FROM JournalEntryLines jel
JOIN Funds f ON jel.FundID = f.FundID
JOIN ChartOfAccounts a ON jel.AccountID = a.AccountID
WHERE a.AccountType IN ('Revenue', 'Expense', 'Other Revenue', 'Other Expense')
AND jel.EntryDate <= '2025-06-30'
GROUP BY
CASE
WHEN f.RestrictionType = 'Unrestricted' THEN 'Without Donor Restrictions'
WHEN f.RestrictionType = 'Temporarily Restricted' THEN 'With Donor Restrictions - Time/Purpose'
WHEN f.RestrictionType = 'Permanently Restricted' THEN 'With Donor Restrictions - Perpetual'
END;

-- Nonprofit Fund Accounting System Query
SELECT
CASE
WHEN f.restriction_type = 'unrestricted' THEN 'Without Donor Restrictions'
WHEN f.restriction_type = 'temporarily_restricted' THEN 'With Donor Restrictions - Time/Purpose'
WHEN f.restriction_type = 'permanently_restricted' THEN 'With Donor Restrictions - Perpetual'
```

```
END AS net_asset_class,
SUM(jel.debit_amount - jel.credit_amount) AS balance
FROM journal_entry_lines jel
JOIN funds f ON jel.fund_id = f.id
JOIN accounts a ON jel.account_id = a.id
JOIN journal_entries je ON jel.journal_entry_id = je.id
WHERE a.type IN ('revenue', 'expense', 'other_revenue', 'other_expense')
AND je.entry_date <= '2025-06-30'
GROUP BY
CASE
WHEN f.restriction_type = 'unrestricted' THEN 'Without Donor Restrictions'
WHEN f.restriction_type = 'temporarily_restricted' THEN 'With Donor Restrictions -
Time/Purpose'
WHEN f.restriction_type = 'permanently_restricted' THEN 'With Donor Restrictions -
Perpetual'
END;
```

## 9.2 Fund Balance Roll-Forward Verification

1Generate Fund Balance Roll-Forward Reports

Generate fund balance roll-forward reports from both systems:

```
-- AccuFund Query
SELECT f.FundCode, f.FundName,
SUM(CASE WHEN jel.EntryDate < '2025-01-01' THEN jel.DebitAmount - jel.CreditAmount
ELSE 0 END) AS BeginningBalance,
SUM(CASE WHEN jel.EntryDate BETWEEN '2025-01-01' AND '2025-06-30' AND
a.AccountType IN ('Revenue', 'Other Revenue') THEN jel.DebitAmount - jel.CreditAmount
ELSE 0 END) AS Revenue,
SUM(CASE WHEN jel.EntryDate BETWEEN '2025-01-01' AND '2025-06-30' AND
a.AccountType IN ('Expense', 'Other Expense') THEN jel.DebitAmount - jel.CreditAmount
ELSE 0 END) AS Expenses,
SUM(CASE WHEN jel.EntryDate <= '2025-06-30' THEN jel.DebitAmount - jel.CreditAmount
ELSE 0 END) AS EndingBalance
FROM Funds f
LEFT JOIN JournalEntryLines jel ON f.FundID = jel.FundID
LEFT JOIN ChartOfAccounts a ON jel.AccountID = a.AccountID
GROUP BY f.FundCode, f.FundName
ORDER BY f.FundCode;

-- Nonprofit Fund Accounting System Query
SELECT f.code AS fund_code, f.name AS fund_name,
SUM(CASE WHEN je.entry_date < '2025-01-01' THEN jel.debit_amount - jel.credit_amount
ELSE 0 END) AS beginning_balance,
SUM(CASE WHEN je.entry_date BETWEEN '2025-01-01' AND '2025-06-30' AND
a.type IN ('revenue', 'other_revenue') THEN jel.debit_amount - jel.credit_amount ELSE
0 END) AS revenue,
SUM(CASE WHEN je.entry_date BETWEEN '2025-01-01' AND '2025-06-30' AND
a.type IN ('expense', 'other_expense') THEN jel.debit_amount - jel.credit_amount ELSE
0 END) AS expenses,
SUM(CASE WHEN je.entry_date <= '2025-06-30' THEN jel.debit_amount - jel.credit_amount ELSE 0 END) AS ending_balance
FROM funds f
LEFT JOIN journal_entry_lines jel ON f.id = jel.fund_id
LEFT JOIN journal_entries je ON jel.journal_entry_id = je.id
LEFT JOIN accounts a ON jel.account_id = a.id
GROUP BY f.code, f.name
ORDER BY f.code;
```

2Compare Fund Balance Roll-Forward

Create a spreadsheet comparing fund balance roll-forward between systems.

## 9.3 Net Asset Release Verification

1Verify Net Asset Releases

Compare net asset releases from restriction between systems:

```
-- AccuFund Query (example - may need customization based on AccuFund setup)
SELECT je.EntryDate, je.Reference, jel.DebitAmount, jel.CreditAmount,
f_from.FundCode AS FromFund, f_to.FundCode AS ToFund
FROM JournalEntries je
JOIN JournalEntryLines jel ON je.JournalEntryID = jel.JournalEntryID
JOIN Funds f_from ON jel.FundID = f_from.FundID
JOIN Funds f_to ON jel.TransferFundID = f_to.FundID
WHERE je.EntryType = 'Net Asset Release'
AND je.EntryDate BETWEEN '2025-01-01' AND '2025-06-30'
ORDER BY je.EntryDate;

-- Nonprofit Fund Accounting System Query
SELECT je.entry_date, je.reference_number, jel.debit_amount, jel.credit_amount,
f_from.code AS from_fund, f_to.code AS to_fund
FROM journal_entries je
JOIN journal_entry_lines jel ON je.id = jel.journal_entry_id
JOIN funds f_from ON jel.fund_id = f_from.id
JOIN funds f_to ON jel.transfer_fund_id = f_to.id
WHERE je.entry_type = 'net_asset_release'
AND je.entry_date BETWEEN '2025-01-01' AND '2025-06-30'
ORDER BY je.entry_date;
```

# 10. Report Comparison Procedures

## 10.1 Standard Financial Reports

1Compare Statement of Financial Position

Generate and compare Balance Sheets (Statement of Financial Position) from both systems:

1. Export reports to Excel from both systems
2. Format reports consistently (remove headers, footers, etc.)
3. Create a comparison spreadsheet with variances
4. Investigate any variances

2Compare Statement of Activities

Generate and compare Income Statements (Statement of Activities) from both systems.

3Compare Statement of Functional Expenses

Generate and compare Statements of Functional Expenses from both systems.

4Compare Statement of Cash Flows

Generate and compare Statements of Cash Flows from both systems.

## 10.2 Fund and Grant Reports

1Compare Fund Reports

Generate and compare fund-specific reports from both systems.

2Compare Grant Reports

Generate and compare grant-specific reports from both systems.

## 10.3 Compliance Reports

1Compare Form 990 Supporting Reports

Generate and compare reports that support Form 990 preparation.

2Compare Grant Compliance Reports

Generate and compare grant compliance reports.

### 10.4 Custom Reports

1Identify Critical Custom Reports

Identify custom reports that are critical to operations.

2Compare Custom Reports

Generate and compare custom reports between systems.

## 11. Variance Analysis and Resolution

### 11.1 Variance Categorization

1Categorize Variances

Categorize identified variances by type:

- **Structural Variances:** Differences in chart of accounts, funds, or entities
- **Transaction Variances:** Missing, duplicate, or incorrect transactions
- **Balance Variances:** Differences in account or fund balances
- **Classification Variances:** Differences in how transactions are classified
- **Timing Variances:** Transactions recorded in different periods
- **Rounding Variances:** Minor differences due to rounding or decimal precision

### 11.2 Variance Investigation

1Investigate Material Variances

For each material variance:

Document the specific variance (account, amount, etc.) Trace back to source transactions in both systems Identify the root cause of the variance Determine if the variance is due to a migration issue