




# TERP Lite — Accounts Module Testing Guide

**Version 5.23.0** · Comprehensive QA checklist for accounting module

**CRITICAL:** This module touches money. Every test must verify both UI behavior AND database integrity. NEVER skip the SQL verification steps.

---

## How to Use

- Tests in **sequence** — many depend on data from earlier tests
  - After each test, mark  Pass /  Fail /  Partial
  - For failures, note exact error message and DB state
  - Use a **TEST tenant** with backup before testing
  - Don't run on production database
- 

## Pre-Test Setup

### Environment

- Fresh test tenant created
- DB backed up
- SQL access to verify `tbl_Ledger`, `tbl_AccountHead`, `tbl_Party`
- Test users:
  - `test_cashier` (Cashier role)
  - `test_acct` (Accountant role)
  - `test_audit` (Auditor role)
  - `test_mgr` (Manager role)
  - `test_party_user` (Party user, linked to test customer)

## Master Data Required

- Currency: LKR (base), USD (foreign)
  - Payment Methods: CASH, CHEQUE, BANK, CARD-VISA
  - Account Groups created (Assets, Liabilities, Income, Expenses, Equity)
  - At least one bank account head ("BOC Account 1234")
  - At least one cash account head ("Cash in Hand")
  - One expense head ("Office Rent")
  - One income head ("Service Income")
- 

## TEST SET 1 — Chart of Accounts

### TC-1.1 Create Account Group

Step	Action	Expected	Result
1	Open Account Groups → New	Form opens	
2	Name = "Operating Expenses", Type = EXPENSE	Saved	
3	Try duplicate name	Error: already exists	

### TC-1.2 Sub-group Hierarchy

Step	Action	Expected	Result
1	New group "Utilities", Parent = Operating Expenses	Saved	
2	Reopen → verify parent shown	Correct	

### TC-1.3 Create Account Head

Step	Action	Expected	Result
1	Open Account Heads → New	OK	
	Code = "5101", Name = "Office Rent", Group =		

2	Operating Expenses	Saved	
3	Try same code again	Error: code exists	

### Verify:

```
SELECT * FROM tbl_AccountHead WHERE AccountCode='5101';
-- Expect 1 row, IsActive=1
```

## TC-1.4 Cannot Delete Used Head

Step	Action	Expected	Result
1	Post a journal using "Office Rent" (jump ahead briefly)	OK	
2	Try Delete on Office Rent	Error: has transactions	
3	Mark IsActive = 0 instead	Saved, hidden in dropdowns	

## TEST SET 2 — Parties

### TC-2.1 Create Customer Party

Step	Action	Expected	Result
1	Open Parties → New	OK	
2	Name = "Test Customer 1", Type = CUSTOMER	Entered	
3	Save	Saved	
4	Verify auto-created account head	New head under Sundry Debtors	

### Verify:

```

SELECT P.PartyID, P.PartyName, AH.AccountName, AG.GroupName
FROM tbl_Party P
JOIN tbl_AccountHead AH ON P.AccountHeadID = AH.AccountHeadID
JOIN tbl_AccountGroup AG ON AH.GroupID = AG.GroupID
WHERE P.PartyName = 'Test Customer 1';
-- Expect group = 'Sundry Debtors' (or similar receivables group)

```

## TC-2.2 Create Supplier Party

Step	Action	Expected	Result
1	New Party, Type = SUPPLIER	Saved	
2	Auto-account under Sundry Creditors	Correct	

## TC-2.3 BOTH Type Party

Step	Action	Expected	Result
1	New Party, Type = BOTH	Saved	
2	Both receivable + payable accounts created	Correct	

## TC-2.4 Party Quick Add

Step	Action	Expected	Result
1	Open Receipt form → "+" next to party	Popup opens	
2	Quick-add party, save	Saved + selected in receipt	

## TC-2.5 Credit Limit

Step	Action	Expected	Result
1	Edit party, Credit Limit = 100,000	Saved	
2	Create transaction exceeding limit	Warning or block (config dependent)	

---

## TEST SET 3 — Opening Balances

### TC-3.1 Opening Party Balance (Debtor)

Step	Action	Expected	Result
1	Open Opening Party Balances → New	OK	
2	Party = Test Customer 1, Date = 1-Apr-2026, Amount = 50,000, DR	Entered	
3	Save	Saved	

#### Verify:

```
SELECT * FROM tbl_Ledger
WHERE RefType IN ('OPENING_BAL','OPENING_PARTY') AND RefID = (select max(...) ...
-- 2 entries: DR Customer, CR Opening Equity (or similar)
-- Both Amount = 50000
```

### TC-3.2 Opening Party Balance (Creditor)

Step	Action	Expected	Result
1	New supplier opening, Amount = 30,000, CR	Saved	
2	GL: CR Supplier, DR Opening Equity	Correct	

### TC-3.3 Opening Bank Balance via Journal

Step	Action	Expected	Result
1	Journal Voucher: DR BOC Account 500,000, CR Capital 500,000	Balanced	
2	Save & Post	Posted	

#### Verify Trial Balance after opening:

```
SELECT
    SUM(CASE WHEN EntryType='DR' THEN Amount ELSE 0 END) AS DR,
    SUM(CASE WHEN EntryType='CR' THEN Amount ELSE 0 END) AS CR
```

```
FROM tbl_Ledger WHERE TenantID = <test>;
-- DR must = CR
```

## TEST SET 4 — Receipts

### TC-4.1 Single Cash Receipt

Step	Action	Expected	Result
1	Login as test_acct	OK	
2	Open Receipt → New	Auto Receipt No assigned	
3	Date = today, Party = Test Customer 1	Selected	
4	Reference = "Cash settlement"	Entered	
5	Add line: CASH, amount = 20,000	Entered	
6	Save	Saved as DRAFT or POSTED depending on config	
7	Commit if needed	Status = POSTED	

#### Verify:

```
SELECT * FROM tbl_Receipt WHERE ReceiptNo LIKE 'RCP%' ORDER BY ReceiptID DESC LIM
SELECT * FROM tbl_ReceiptDetail WHERE ReceiptID = (above);
SELECT * FROM tbl_Ledger WHERE RefType='RECEIPT' AND RefID = (above);
-- 2 ledger entries: DR Cash, CR Customer
-- Both Amount = 20000
```

### TC-4.2 Multi-Method Receipt

Step	Action	Expected	Result
1	New Receipt, Party = Test Customer 1	OK	
2	Line 1: CASH 10,000	Entered	

3	Line 2: CHEQUE, ref CH-100, 15,000	Entered	
4	Line 3: BANK, ref TXN-200, 5,000	Entered	
5	Total = 30,000, Save & Commit	OK	

### Verify:

```
SELECT * FROM tbl_ReceiptDetail WHERE ReceiptID=?;
-- 3 rows, amounts 10000+15000+5000=30000
```

### TC-4.3 Cheque Status Update

Step	Action	Expected	Result
1	Open Cheque Register	Cheque CH-100 listed as Pending	
2	Mark CH-100 as Cleared (date = today)	Status changes	
3	Cheque CH-101 mark as Bounced	Status changes, reversal journal posted	

### Verify bounced cheque reverses entries:

```
SELECT * FROM tbl_Ledger WHERE Narration LIKE '%bounced%';
-- Should have reversing entries
```

### TC-4.4 Receipt Reverses Customer Balance

Step	Action	Expected	Result
1	Customer opening = 50,000 (TC-3.1)	OK	
2	Receipts = 20,000 (TC-4.1) + 30,000 (TC-4.2)	OK	
3	Open Ledger Statement → Test Customer 1	Closing balance = 0	

## TEST SET 5 — Payments

## TC-5.1 Office Rent Payment

Step	Action	Expected	Result
1	Open Payment → New	OK	
2	Date, Party = Landlord (create if needed)	Entered	
3	Line 1: Account = Office Rent, Amount = 30,000	Entered	
4	Payment Method = BANK, ref = TXN-001	Entered	
5	Save & Commit	OK	

### Verify:

```
SELECT * FROM tbl_Ledger WHERE RefType='PAYMENT' AND RefID=(latest)
-- DR Office Rent 30,000
-- CR BOC Account 30,000
```

## TC-5.2 Supplier Payment

Step	Action	Expected	Result
1	New Payment, Party = Test Supplier (TC-2.2)	OK	
2	Account = Test Supplier's account, Amount = 30,000	Entered	
3	Pay via CHEQUE, ref = CH-500	Entered	
4	Save & Commit	OK	

### Verify supplier balance is now 0:

```
-- Was 30000 from TC-3.2, paid 30000, balance = 0
SELECT
    SUM(CASE WHEN EntryType='CR' THEN Amount ELSE -Amount END) AS Balance
FROM tbl_Ledger
WHERE AccountHeadID = (SELECT AccountHeadID FROM tbl_Party WHERE PartyName='Test')
-- Expect 0
```

## TC-5.3 Multi-Line Expense Payment

--	--	--	--



Step	Action	Expected	Result
1	New Payment	OK	
2	Line 1: Salaries 200,000	OK	
3	Line 2: EPF 24,000	OK	
4	Line 3: ETF 6,000	OK	
5	Pay via Bank, total 230,000	OK	

### Verify:

```
SELECT COUNT(*) FROM tbl_PaymentDetail WHERE PaymentID=?;
-- 3 lines
```

---

## TEST SET 6 — Journal Vouchers

### TC-6.1 Balanced Journal

Step	Action	Expected	Result
1	Open Journal Voucher → New	OK	
2	Date, Narration = "Test depreciation"	OK	
3	Line 1: DR Depreciation Expense 10,000	OK	
4	Line 2: CR Accumulated Depreciation 10,000	OK	
5	Save & Post	OK	

### TC-6.2 Unbalanced Rejection

Step	Action	Expected	Result
1	New Journal	OK	
2	DR 5,000, CR 4,000	Entered	
3	Try Save	Error: not balanced	

### TC-6.3 Multi-Line Journal

Step	Action	Expected	Result
1	4-line adjustment entry	All balanced	
2	Save	OK	

### TC-6.4 Cost Center / Branch Tagging

Step	Action	Expected	Result
1	Journal line with BranchID = Branch1	OK	
2	Run report filter by Branch1	Entry appears	

## TEST SET 7 — Income/Expense Quick Entry

### TC-7.1 Quick Income

Step	Action	Expected	Result
1	Open Income/Expense Entry	OK	
2	Type = INCOME, Head = Service Income, Amount = 5,000	OK	
3	Received Via = CASH	OK	
4	Save	Posted automatically	

**Verify:** 2 ledger entries (DR Cash 5000, CR Service Income 5000).

### TC-7.2 Quick Expense

Step	Action	Expected	Result
1	Type = EXPENSE, Head = Stationery 2,500	OK	
2	Paid Via = CASH	OK	

3	Save	Posted	
---	------	--------	--

## TEST SET 8 — Recurring Journals

### TC-8.1 Define Recurring

Step	Action	Expected	Result
1	Open Recurring Journals → New	OK	
2	Name = "Monthly Rent"	OK	
3	Frequency = MONTHLY, NextRunDate = today	OK	
4	Auto-post = YES	OK	
5	Line 1: DR Office Rent 30,000	OK	
6	Line 2: CR BOC Account 30,000	OK	
7	Save	OK	

### TC-8.2 Auto Post at Login

Step	Action	Expected	Result
1	Logout, login again	OK	
2	Check Voucher List	New journal auto-posted today	

#### Verify:

```
SELECT * FROM tbl_RecurringJournal WHERE RecurringID=?;
-- NextRunDate advanced by 1 month
```

```
SELECT * FROM tbl_Ledger WHERE Narration LIKE '%Monthly Rent%' ORDER BY EntryID D
-- Auto entries present
```

### TC-8.3 Manual Run

--	--	--	--

Step	Action	Expected	Result
1	Open recurring, click "Run Now"	Confirmation	
2	Confirm	New journal posted	

#### TC-8.4 End Date Honored

Step	Action	Expected	Result
1	Set EndDate = yesterday	OK	
2	Try Run Now	Error or skip	

### TEST SET 9 — Bank Book & Reconciliation

#### TC-9.1 Bank Book View

Step	Action	Expected	Result
1	Open Bank Book	OK	
2	Select BOC Account	All BOC transactions listed	
3	Date filter last 30 days	Filtered	
4	Running balance correct at each line	Correct	

#### TC-9.2 Reconciliation Process

Step	Action	Expected	Result
1	Open Bank Reconciliation	OK	
2	Bank = BOC, Statement Date = today	OK	
3	Enter Statement Balance = 450,000	OK	
4	System lists 5 transactions	OK	
5	Tick 4 as cleared, leave 1 unticked (uncleared cheque)	OK	

6	System shows reconciled balance	OK	
7	Difference = 0 or matches uncleared amount	Correct	
8	Save reconciliation	OK	

### Verify:

```
SELECT IsReconciled, ReconciliationID FROM tbl_Ledger
WHERE AccountHeadID = <BOC> AND EntryDate BETWEEN <range>;
-- 4 marked reconciled, 1 not
```

## TEST SET 10 — Fixed Assets

### TC-10.1 Add Asset

Step	Action	Expected	Result
1	Open Fixed Assets → New	OK	
2	Name = "Office Computer Dell", Cost = 80,000	OK	
3	Useful Life = 5 years, Method = Straight-line	OK	
4	Salvage = 5,000	OK	
5	Save	Asset code auto-generated	

### TC-10.2 Depreciation Recurring

Step	Action	Expected	Result
1	Create recurring journal: monthly depreciation $(80000 - 5000) / 60 = 1,250$	OK	
2	DR Depreciation 1,250 / CR Accumulated Dep 1,250	OK	
3	Run	Posted	

## TC-10.3 Asset Disposal

Step	Action	Expected	Result
1	Mark asset = Disposed, Sale Price = 50,000	OK	
2	Manual journal: complex (cost, accumulated dep, sale, gain/loss)	OK	

## TEST SET 11 — Period Close

### TC-11.1 Lock Period

Step	Action	Expected	Result
1	Login as test_mgr	OK	
2	Open Period Close	OK	
3	Select previous month	OK	
4	Click Lock	Confirmation	
5	Status = LOCKED	OK	

### TC-11.2 Cannot Post in Locked Period

Step	Action	Expected	Result
1	New Receipt, date = last month	Entered	
2	Try save	Error: period locked	

### TC-11.3 Manager Unlock

Step	Action	Expected	Result
1	As Manager, unlock previous month	OK	
2	Audit log entry created	OK	

3	Now can post	OK	
---	--------------	----	--

### Verify:

```
SELECT * FROM tbl_PeriodLock ORDER BY PeriodLockID DESC LIMIT 5;
-- Lock/unlock history visible
```

```
SELECT * FROM tbl_AuditLog WHERE TableName='tbl_PeriodLock' ORDER BY AuditID DESC
-- Manager unlock logged
```

## TEST SET 12 — Reports

### TC-12.1 Trial Balance Balanced

Step	Action	Expected	Result
1	Open Trial Balance	OK	
2	As-of-date = today	OK	
3	Total Debits = Total Credits	EQUAL	
4	Each account shows correct balance	Spot check 3 accounts	

### Critical verification:

```
SELECT
    SUM(CASE WHEN EntryType='DR' THEN Amount ELSE 0 END) AS DR,
    SUM(CASE WHEN EntryType='CR' THEN Amount ELSE 0 END) AS CR
FROM tbl_Ledger
WHERE TenantID = <test> AND EntryDate <= GETDATE();
-- DR MUST = CR (down to the cent)
```

### TC-12.2 Ledger Statement

Step	Action	Expected	Result
1	Open Ledger Statement	OK	
2	Select Test Customer 1, period = current month	OK	

3	Opening balance + transactions = Closing	Math works	
4	Export to Excel	OK	

### TC-12.3 P&L Statement

Step	Action	Expected	Result
1	Open P&L Statement	OK	
2	Period = current FY	OK	
3	Income > 0, Expenses > 0	Both > 0 (from above tests)	
4	Net Profit/Loss displayed	Calculated correctly	
5	Drill down on any expense → see ledger	Works	

### TC-12.4 Balance Sheet

Step	Action	Expected	Result
1	Open Balance Sheet	OK	
2	As-of-date = today	OK	
3	Assets = Liabilities + Equity	EQUAL	
4	Print/Export	OK	

### TC-12.5 Cash Flow

Step	Action	Expected	Result
1	Open Cash Flow Statement	OK	
2	Period = current month	OK	
3	Opening Cash + Net Change = Closing Cash	Math works	
4	Operating + Investing + Financing = Net	Components sum	

### TC-12.6 Customer Aging



Step	Action	Expected	Result
1	Create receivable invoice (or use opening balance still unpaid)	OK	
2	Open Customer Aging	OK	
3	Customer appears in correct bucket (Current/31-60/etc.)	Correct	
4	Total of all customers = Sundry Debtors balance	Match	

### TC-12.7 Supplier Aging

Step	Action	Expected	Result
1	Open Supplier Aging	OK	
2	Bucketed correctly	Correct	
3	Total = Sundry Creditors balance	Match	

### TC-12.8 Cheque Register

Step	Action	Expected	Result
1	Open Cheque Register	OK	
2	Filter status = Pending	Shows pending only	
3	Update status, bank charge entry created	Correct	

## TEST SET 13 — Dashboard

### TC-13.1 Accounts Dashboard KPIs

Step	Action	Expected	Result
1	Open Accounts Dashboard	Loads	
2	Cash balance correct	Matches Trial Balance	
3	Bank balance correct	Matches	

4	Receivable balance correct	Matches Customer Aging total	
5	Payable balance correct	Matches Supplier Aging total	
6	Charts render	OK	

## TC-13.2 Performance with Volume

Generate 1,000 ledger entries via script, then:

Step	Action	Expected	Result
1	Open Dashboard	<5 seconds	
2	Open Trial Balance	<10 seconds	
3	Drill-down works	OK	

## TEST SET 14 — Party Portal

### TC-14.1 Setup Portal Access

Step	Action	Expected	Result
1	Open Party Users → New	OK	
2	Party = Test Customer 1, username = customer1, password = TestPass	Saved	
3	Assign Role = "Owner"	OK	

### TC-14.2 Party Login

Step	Action	Expected	Result
1	Logout, login as customer1	OK	
2	See Party Portal menus only	OK	
3	Cannot see internal menus	Correct	

### TC-14.3 Party Statement

Step	Action	Expected	Result
1	Open My Statement	Shows own transactions only	
2	Cannot see other customers' data	Confirmed	
3	Running balance correct	Matches admin view	

### TC-14.4 Submit Request

Step	Action	Expected	Result
1	Open New Request	OK	
2	Type = Refund Request, Amount = 1,000	OK	
3	Save	Submitted	

### TC-14.5 Admin Sees Request

Step	Action	Expected	Result
1	Login as Manager	OK	
2	Open Incoming Requests	New request visible	
3	Approve	Status updates	
4	Customer sees status update on next login	OK	

---

## TEST SET 15 — Permissions

### TC-15.1 Cashier Restrictions

Step	Action	Expected	Result
1	Login as test_cashier	OK	
2	Can access Receipt	OK	

3	Cannot access Payment	Hidden or Denied	
4	Cannot access Journal Voucher	Hidden	
5	Cannot delete master data	Hidden	

### TC-15.2 Auditor Read-Only

Step	Action	Expected	Result
1	Login as test_audit	OK	
2	View Trial Balance, P&L	OK	
3	Try edit any transaction	Read-only	
4	Cannot save new transactions	Blocked	

### TC-15.3 Manager Full Access

Step	Action	Expected	Result
1	Login as test_mgr	OK	
2	Can perform every action	OK	
3	Can lock/unlock periods	OK	

## TEST SET 16 — Multi-Currency

### TC-16.1 USD Receipt

Step	Action	Expected	Result
1	Receipt in USD, FX = 320, Amount = 100 USD	OK	
2	Save	OK	

**Verify:** GL posted in LKR (32,000) with USD/FX captured.

### TC-16.2 Multi-Currency Trial Balance

Step	Action	Expected	Result
1	Trial balance in LKR (base)	OK	
2	All foreign transactions converted	At correct FX rate	

## TEST SET 17 — Edge Cases

### TC-17.1 Zero Amount

Step	Action	Expected	Result
1	Try receipt with amount = 0	Validation error	

### TC-17.2 Negative Amount

Step	Action	Expected	Result
1	Try -100 amount	Validation error	

### TC-17.3 Future Date

Step	Action	Expected	Result
1	Receipt dated next year	Warning or allowed	

### TC-17.4 Past Date in Locked Period

Step	Action	Expected	Result
1	Try date in locked month	Error: period locked	

### TC-17.5 Concurrent Edit

Step	Action	Expected	Result
1	User A opens receipt	OK	
2	User B opens same receipt	OK	

3	A saves, B saves	B gets error or merge prompt	
---	------------------	------------------------------	--

## TC-17.6 Special Characters in Narration

Step	Action	Expected	Result
1	Narration with ', ", &	Saved correctly (no SQL injection)	

## TC-17.7 Long Narration

Step	Action	Expected	Result
1	1000-char narration	Truncated or saved	

# TEST SET 18 — GL Integrity (CRITICAL)

After all tests, run these checks:

## TC-18.1 Total Debits = Total Credits

```
SELECT
    SUM(CASE WHEN EntryType='DR' THEN Amount ELSE 0 END) AS TotalDR,
    SUM(CASE WHEN EntryType='CR' THEN Amount ELSE 0 END) AS TotalCR,
    SUM(CASE WHEN EntryType='DR' THEN Amount ELSE -Amount END) AS Diff
FROM tbl_Ledger
WHERE TenantID = <test>;
```

**Must:** TotalDR = TotalCR, Diff = 0

| Result: | |

## TC-18.2 Each Transaction Self-Balanced

```
SELECT RefType, RefID,
    SUM(CASE WHEN EntryType='DR' THEN Amount ELSE -Amount END) AS Diff
FROM tbl_Ledger
WHERE TenantID = <test>
GROUP BY RefType, RefID
HAVING SUM(CASE WHEN EntryType='DR' THEN Amount ELSE -Amount END) <> 0;
```

**Must:** Return 0 rows (every txn balances on its own)

| Result: ||

### TC-18.3 Trial Balance = Reports

- Sum of asset accounts = Balance Sheet Assets
- Sum of income - expenses = P&L Net Profit
- Cash + Bank from Trial Balance = Cash Flow closing
- Sundry Debtors = sum of all positive Customer Aging
- Sundry Creditors = sum of all positive Supplier Aging

| All consistent? | Yes / No |

### TC-18.4 No Orphan Ledger Entries

```
SELECT * FROM tbl_Ledger
WHERE AccountHeadID NOT IN (SELECT AccountHeadID FROM tbl_AccountHead)
    OR (PartyID IS NOT NULL AND PartyID NOT IN (SELECT PartyID FROM tbl_Party));
```

**Must:** 0 rows

| Result: ||

---

## TEST SET 19 — Cleanup

```
-- WARNING: DEV ONLY, deletes all test data
DELETE FROM tbl_PaymentDetailAttribute WHERE PaymentDetailID IN (SELECT PaymentDe
DELETE FROM tbl_PaymentDetail WHERE PaymentID IN (SELECT PaymentID FROM tbl_Payme
DELETE FROM tbl_Payment WHERE TenantID = <test>;
DELETE FROM tbl_ReceiptDetailAttribute WHERE ReceiptDetailID IN (...);
DELETE FROM tbl_ReceiptDetail WHERE ReceiptID IN (SELECT ReceiptID FROM tbl_Recei
DELETE FROM tbl_Receipt WHERE TenantID = <test>;
DELETE FROM tbl_Ledger WHERE TenantID = <test>;
DELETE FROM tbl_AccountHead WHERE TenantID = <test>;
DELETE FROM tbl_AccountGroup WHERE TenantID = <test>;
DELETE FROM tbl_Party WHERE TenantID = <test>;
DELETE FROM tbl_PeriodLock WHERE TenantID = <test>;
DELETE FROM tbl_RecurringJournal WHERE TenantID = <test>;
```

Or just use a fresh test tenant.

## Sign-Off

Test Set	Total	Pass	Fail	Tester	Date	Notes
1. Chart of Accounts	4					
2. Parties	5					
3. Opening Balances	3					
4. Receipts	4					
5. Payments	3					
6. Journal Vouchers	4					
7. Income/Expense	2					
8. Recurring Journals	4					
9. Bank Recon	2					
10. Fixed Assets	3					
11. Period Close	3					
12. Reports	8					
13. Dashboard	2					
14. Party Portal	5					
15. Permissions	3					
16. Multi-currency	2					
17. Edge cases	7					
18. GL Integrity	4					

**Overall Status:** ☐ PASS ☐ FAIL ☐ CONDITIONAL

**Approved for Production:** \_\_\_\_\_



**Date:** \_\_\_\_\_

---

**REMEMBER:** This module handles real money. A passing test that's not been verified at the DB level isn't actually passing. **Always run the SQL verification queries.**

**End of Testing Guide** Version 5.23.0