**SIA Extract System - Test Case Document (TCD)**

**Document Control**

* **Application**: SIA Extract (Java Implementation - Migrated from siaextract.pc)
* **Original System**: Oracle Pro\*C Legacy Application
* **Current Platform**: Java-based Database Extraction Utility
* **Document Version**: 1.0
* **Date**: September 29, 2025
* **Prepared By**: QA Engineering Team
* **Approval**: Pending

**Table of Contents**

1. [Executive Summary](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#executive-summary)
2. [Test Environment Setup](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#test-environment-setup)
3. [Test Data Requirements](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#test-data-requirements)
4. [Functional Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#functional-test-cases)
5. [Negative Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#negative-test-cases)
6. [Boundary Value Analysis (BVA)](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#boundary-value-analysis)
7. [Integration Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#integration-test-cases)
8. [Performance Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#performance-test-cases)
9. [Security Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#security-test-cases)
10. [Data Integrity Test Cases](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#data-integrity-test-cases)
11. [Traceability Matrix](https://claude.ai/chat/939831e5-a013-4d9c-89e4-f859efda9705#traceability-matrix)

**1. Executive Summary**

**1.1 Purpose**

This Test Case Document provides comprehensive testing specifications for the Java implementation of the SIA Extract system, originally developed in Oracle Pro\*C. The document ensures functional equivalence, data integrity, and performance parity between legacy and migrated systems.

**1.2 Application Overview**

The SIA Extract application extracts taxpayer information records from Oracle database tables (ALS.ASSN and DIAL.COREDIAL) and generates formatted output files for downstream processing. The system handles three transaction types:

* **TSIGN**: Standard transactions (no area/SC change)
* **TC015**: Transactions requiring area change but not SC change
* **TC030**: Transactions requiring both area and SC change

**1.3 Key Business Rules**

1. Service Center codes must be in approved list: 7, 8, 9, 17, 18, 19, 28, 29, 49, 89
2. UPDTFLAG controls database modification (Y=update, N=read-only)
3. Transaction eligibility determined by trans\_mod value (0=skip, 1=TSIGN, 2=TC015, 3=TC030)
4. Financial System codes: 1=IMF, 2=BMF, 3=EPEF, 4=IRAF, 6=NMF
5. Only NMF (fs=6) requires ULC Code upload
6. Posting delay code "1" appended to TC015/TC030 to prevent weekend processing issues

**2. Test Environment Setup**

**2.1 Prerequisites**

**Database Environment:**

* Oracle Database 11g or higher
* ALS schema with ASSN table accessible
* DIAL schema with COREDIAL table accessible
* DOMAP reference table populated
* Test database isolated from production

**Application Environment:**

* Java Runtime Environment (JRE) 8 or higher
* JDBC driver for Oracle connectivity
* Application JAR file: sia-extract.jar
* Configuration file: sia-config.properties

**File System:**

* Directory structure: SC##/ directories for output (where ## = 07, 08, 09, 17, 18, 19, 28, 29, 49, 89)
* Write permissions on output directories
* Error log directory with write permissions

**Test Tools:**

* SQL\*Plus or SQL Developer for database verification
* Text editor for output file inspection
* Log analysis tools

**2.2 Test User Credentials**

* **Database User**: test\_als\_user
* **Password**: Encrypted via application configuration
* **Privileges**: SELECT on ALS.ASSN, DIAL.COREDIAL, DOMAP; UPDATE on ALS.ASSN

**3. Test Data Requirements**

**3.1 Minimum Test Data Set**

**TIN Records by Category:**

* At least 50 records per Service Center (SC 7, 8, 9, 17, 18, 19, 28, 29, 49, 89)
* Records with LOADDT = NULL or '01/01/1900' (NULLDATE)
* Records with specific test dates (e.g., current date, future date, past date)

**Transaction Type Distribution:**

* trans\_mod = 0 (not eligible): 10% of records
* trans\_mod = 1 (TSIGN): 50% of records
* trans\_mod = 2 (TC015): 20% of records
* trans\_mod = 3 (TC030): 20% of records

**Financial System Coverage:**

* fs = 1 (IMF): 30% of records
* fs = 2 (BMF): 25% of records
* fs = 3 (EPEF): 15% of records
* fs = 4 (IRAF): 10% of records
* fs = 6 (NMF): 20% of records

**Status and Pick Indicators:**

* STATUS = 'O' or 'Q'
* QPICKIND = 3 (eligible for processing)
* QPICKIND = 4 (already processed - for negative tests)

**3.2 Test Data Setup Scripts**

-- Create test TINs for SC 07

INSERT INTO ALS.ASSN (TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY,

GRNUM, RONUM, NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES (100000001, 1, 4, '01/01/1900', 'O', 3, 5, 7, 12, 3, 'TEST', NULL,

SYSDATE, 1, 1);

-- Create test records in DIAL.COREDIAL

INSERT INTO DIAL.COREDIAL (TIN, FS, TT, CORETIN, COREFS, CORETT)

VALUES (100000001, 1, 4, 100000001, 1, 4);

-- Repeat for various scenarios

**4. Functional Test Cases**

**TC\_SIA\_F\_001: Display Application Version**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_001 |
| **Test Case Name** | Display Application Version |
| **Test Type** | Functional - Positive |
| **Priority** | Medium |
| **Module** | Command Line Interface |

**Prerequisites:**

* Application deployed and executable
* Java runtime configured

**Test Steps:**

1. Open command prompt/terminal
2. Navigate to application directory
3. Execute command: java -jar sia-extract.jar -version
4. Observe console output

**Expected Results:**

* Application displays version number
* Application displays build date or cookanbool date
* Application exits with status code 0
* No error messages displayed
* No database connection attempted

**Test Data:** None required

**Pass/Fail Criteria:**

* ✓ PASS: Version information displayed correctly and application exits cleanly
* ✗ FAIL: Error thrown, incorrect version, or application hangs

**Notes:** Verify version matches deployment documentation

**TC\_SIA\_F\_002: Execute with Valid Parameters - Single SC**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_002 |
| **Test Case Name** | Execute with Valid Parameters for Single Service Center |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Data Extraction |

**Prerequisites:**

* Database accessible with valid credentials
* Test data exists for SC 07 with LOADDT '09/29/2025'
* Output directory SC07/ exists with write permissions

**Test Steps:**

1. Execute command: java -jar sia-extract.jar -fix 07 09/29/2025 Y
2. Monitor console for status messages
3. Verify database connection established
4. Wait for completion message
5. Navigate to SC07/ directory
6. Verify output file created

**Expected Results:**

* Application connects to database successfully
* Message displayed: "Processing SC 07..."
* Records extracted from ALS.ASSN table
* Output file created: SC07/alsent.07.20250929
* File contains formatted records
* LOADDT in database updated to system date
* QPICKIND updated from 3 to 4
* Application exits with status code 0
* Console shows: "Processing completed. X records extracted."

**Test Data:**

* SC: 07
* LOADDT: 09/29/2025
* UPDTFLAG: Y
* Expected TIN count: At least 10 records

**Pass/Fail Criteria:**

* ✓ PASS: File created with correct naming, records properly formatted, database updated
* ✗ FAIL: File not created, incorrect format, database not updated, or application error

**TC\_SIA\_F\_003: Execute with SC=0 (All Service Centers)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_003 |
| **Test Case Name** | Extract All Service Centers Simultaneously |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Data Extraction |

**Prerequisites:**

* Database contains records for multiple SCs (at minimum: 7, 8, 9, 17)
* All SC##/ directories exist
* Test date has data across multiple SCs

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 0 09/29/2025 Y
2. Monitor processing for all SC codes
3. Verify console shows processing for each SC
4. Check all output directories

**Expected Results:**

* Application processes all valid SC codes: 7, 8, 9, 17, 18, 19, 28, 29, 49, 89
* Multiple output files created:
  + SC07/alsent.07.20250929
  + SC08/alsent.08.20250929
  + SC09/alsent.09.20250929
  + SC17/alsent.17.20250929
  + (etc. for SCs with data)
* Console displays record count for each SC
* For SCs without data: "No records found for SC XX"
* All database updates applied
* Application completes successfully

**Test Data:**

* SC: 0 (all)
* LOADDT: 09/29/2025
* UPDTFLAG: Y
* Expected: Multiple SCs have records

**Pass/Fail Criteria:**

* ✓ PASS: All SC files created correctly, counts accurate, database updated
* ✗ FAIL: Missing files, incorrect counts, processing stops prematurely

**TC\_SIA\_F\_004: UPDTFLAG=Y - Database Update Verification**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_004 |
| **Test Case Name** | Verify Database Updates When UPDTFLAG=Y |
| **Test Type** | Functional - Positive |
| **Priority** | Critical |
| **Module** | Database Update |

**Prerequisites:**

* Known test TIN: 100000001
* Original LOADDT: '09/25/2025'
* Original QPICKIND: 3
* Original TRANSMOD: NULL

**Test Steps:**

1. Query original values:
2. SELECT TIN, LOADDT, TRANSMOD, QPICKIND FROM ALS.ASSN WHERE TIN = 100000001;
3. Record values: LOADDT='09/25/2025', QPICKIND=3, TRANSMOD=NULL
4. Execute: java -jar sia-extract.jar -fix 07 09/25/2025 Y
5. Re-query same TIN
6. Verify LOADDT updated to current system date
7. Verify QPICKIND updated to 4
8. Verify TRANSMOD updated to appropriate trans\_mod value

**Expected Results:**

* **Before Extraction:**
  + LOADDT = '09/25/2025'
  + QPICKIND = 3
  + TRANSMOD = NULL
* **After Extraction:**
  + LOADDT = TO\_CHAR(SYSDATE, 'MM/DD/YYYY') (current date)
  + QPICKIND = 4
  + TRANSMOD = 1 (or appropriate trans\_mod value based on data)
* SQL UPDATE statements executed successfully
* No database errors in log
* Changes committed to database

**Test Data:**

* Test TIN: 100000001
* SC: 07
* LOADDT: 09/25/2025
* UPDTFLAG: Y

**Pass/Fail Criteria:**

* ✓ PASS: All three fields updated correctly
* ✗ FAIL: Any field remains unchanged or incorrect value

**TC\_SIA\_F\_005: UPDTFLAG=N - No Database Modification**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_005 |
| **Test Case Name** | Verify No Database Updates When UPDTFLAG=N |
| **Test Type** | Functional - Positive |
| **Priority** | Critical |
| **Module** | Database Update |

**Prerequisites:**

* Known test TIN: 100000002
* Original values documented

**Test Steps:**

1. Query and record original values:
2. SELECT TIN, LOADDT, TRANSMOD, QPICKIND, ROWIDFROM ALS.ASSN WHERE TIN = 100000002;
3. Execute: java -jar sia-extract.jar -fix 07 09/25/2025 N
4. Verify output file created with record for TIN 100000002
5. Re-query database
6. Compare all field values

**Expected Results:**

* Output file contains extracted record
* **Database values UNCHANGED:**
  + LOADDT = original value
  + QPICKIND = 3 (unchanged)
  + TRANSMOD = original value
  + ROWID = same
* No UPDATE SQL statements executed
* Console may show: "Running in read-only mode (UPDTFLAG=N)"
* Application completes successfully

**Test Data:**

* Test TIN: 100000002
* SC: 07
* LOADDT: 09/25/2025
* UPDTFLAG: N

**Pass/Fail Criteria:**

* ✓ PASS: File created, record extracted, database unchanged
* ✗ FAIL: Any database field modified

**Notes:** This mode used for reporting/auditing without altering production data

**TC\_SIA\_F\_006: Transaction Type TSIGN (trans\_mod=1)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_006 |
| **Test Case Name** | Verify TSIGN Record Format Generation |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Record Formatting |

**Prerequisites:**

* Test TIN: 100000010
* trans\_mod = 1 (eligible for TSIGN)
* fs = 1, tt = 4
* nmctrl = 'AB12', ronum = '3456', aoto = '7890'

**Test Steps:**

1. Execute extraction: java -jar sia-extract.jar -fix 07 09/29/2025 Y
2. Open output file: SC07/alsent.07.20250929
3. Locate record for TIN 100000010
4. Verify record format character by character

**Expected Results:**

* Record format: 07620100000010140000000000000U0TSIGNAB123456789000000
* **Field breakdown:**
  + Positions 1-2: 07 (SC code)
  + Positions 3-5: 620 (constant)
  + Positions 6-14: 100000010 (TIN, 9 digits)
  + Position 15: 1 (FS)
  + Position 16: 4 (TT)
  + Positions 17-28: 000000000000 (12 zeros)
  + Positions 29-30: U0 (constant)
  + Positions 31-35: TSIGN (transaction type)
  + Positions 36-39: AB12 (nmctrl, 4 chars)
  + Positions 40-43: 3456 (ronum, 4 chars)
  + Positions 44-47: 7890 (aoto, 4 chars)
  + Positions 48-53: 000000 (6 chars padding)
  + Position 54: 0 (constant)
  + Position 55: newline

**Test Data:**

* TIN: 100000010
* trans\_mod: 1
* fs: 1
* tt: 4
* nmctrl: 'AB12'
* ronum: '3456'
* aoto: '7890'

**Pass/Fail Criteria:**

* ✓ PASS: Record exactly matches expected format
* ✗ FAIL: Any field misaligned, missing, or incorrect value

**TC\_SIA\_F\_007: Transaction Type TC015/IFEUP (trans\_mod=2, fs=1)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_007 |
| **Test Case Name** | Verify TC015 IFEUP Record Format (IMF) |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Record Formatting |

**Prerequisites:**

* Test TIN: 100000020
* trans\_mod = 2 (requires TC015)
* fs = 1 (IMF)
* nmctrl = 'XY34', aoto = '1234', ronum = '5678'

**Test Steps:**

1. Execute extraction
2. Open output file
3. Locate record for TIN 100000020
4. Verify IFEUP015 format with posting delay code

**Expected Results:**

* Record format includes:
  + IFEUP015 indicator (not BFEUP)
  + First 2 characters of aoto for CLC: 12
  + Posting delay code 1 at end
* Full format: 07620100000020140000000000000U0IFEUP015XY3450012125678 1
* **Key elements:**
  + Transaction type: IFEUP015
  + Uses 500 + first 2 chars of aoto for CLC
  + Ends with posting delay: 1

**Test Data:**

* TIN: 100000020
* trans\_mod: 2
* fs: 1
* aoto: '1234'

**Pass/Fail Criteria:**

* ✓ PASS: IFEUP015 format correct, posting delay '1' present
* ✗ FAIL: Wrong transaction type, missing posting delay, or format error

**Notes:** Posting delay "1" critical to prevent weekend processing timing issues

**TC\_SIA\_F\_008: Transaction Type TC015/BFEUP (trans\_mod=2, fs=6, NMF with ULC)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_008 |
| **Test Case Name** | Verify TC015 BFEUP Record Format for NMF with ULC Code |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Record Formatting |

**Prerequisites:**

* Test TIN: 100000030
* trans\_mod = 2
* fs = 6 (NMF - requires ULC)
* ulccd = 25
* nmctrl = 'CD56', aoto = '9012', ronum = '3456'

**Test Steps:**

1. Execute extraction
2. Open output file
3. Locate record for TIN 100000030
4. Verify ULC code included in output

**Expected Results:**

* Record format: 07620100000030640000000000000U0BFEUP015CD565025903456 1
* **Key differences from non-NMF:**
  + Transaction type: BFEUP015
  + ULC code included: 25 (2-digit format)
  + Format: 50{ULC} instead of 500
  + Posting delay: 1
* **Business Rule Validated**: Only NMF (fs=6) includes ULC\_CD

**Test Data:**

* TIN: 100000030
* trans\_mod: 2
* fs: 6
* ulccd: 25

**Pass/Fail Criteria:**

* ✓ PASS: BFEUP015 format with ULC code '25' present
* ✗ FAIL: ULC missing, incorrect format, or wrong transaction type

**TC\_SIA\_F\_009: Transaction Type TC030 (trans\_mod=3)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_009 |
| **Test Case Name** | Verify TC030 Record Format (Area and SC Change) |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Record Formatting |

**Prerequisites:**

* Test TIN: 100000040
* trans\_mod = 3 (requires TC030)
* fs = 2 (BMF)

**Test Steps:**

1. Execute extraction
2. Verify output contains TC030 indicator
3. Confirm posting delay code '1'

**Expected Results:**

* Transaction indicator: BFEUP030 (or IFEUP030 if IMF/IRAF)
* Format similar to TC015 but with 030 instead of 015
* Posting delay code 1 present
* **Business Rule**: Indicates both area AND service center change

**Test Data:**

* TIN: 100000040
* trans\_mod: 3
* fs: 2

**Pass/Fail Criteria:**

* ✓ PASS: Correct TC030 format with posting delay
* ✗ FAIL: Wrong transaction indicator or missing posting delay

**TC\_SIA\_F\_010: Skip Non-Eligible Records (trans\_mod=0)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_F\_010 |
| **Test Case Name** | Verify Non-Eligible Records Skipped |
| **Test Type** | Functional - Positive |
| **Priority** | High |
| **Module** | Record Filtering |

**Prerequisites:**

* Test TIN: 100000050
* trans\_mod = 0 (not eligible)
* Record exists in database and retrieved in hook query

**Test Steps:**

1. Query database to confirm TIN 100000050 has trans\_mod = 0
2. Execute extraction
3. Verify TIN retrieved in initial query (check debug logs if available)
4. Open output file
5. Search for TIN 100000050

**Expected Results:**

* TIN 100000050 retrieved in TIN hook query
* Record processed in detail loop
* trans\_mod = 0 detected
* Record **NOT written** to output file
* No error generated
* Processing continues to next TIN
* Database **NOT updated** for this TIN (LOADDT and QPICKIND unchanged)

**Test Data:**

* TIN: 100000050
* trans\_mod: 0

**Pass/Fail Criteria:**

* ✓ PASS: TIN not in output file, processing continues normally
* ✗ FAIL: TIN appears in output file or processing errors

**5. Negative Test Cases**

**TC\_SIA\_N\_001: Invalid Service Center Code**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_001 |
| **Test Case Name** | Invalid SC Code Rejection |
| **Test Type** | Functional - Negative |
| **Priority** | High |
| **Module** | Input Validation |

**Prerequisites:**

* None

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 99 09/29/2025 Y
2. Observe error message
3. Verify application termination

**Expected Results:**

* Application validates SC code against approved list [7,8,9,17,18,19,28,29,49,89]
* SC 99 not in approved list
* Error message displayed: "Invalid SC Code: 99. Valid codes are: 7, 8, 9, 17, 18, 19, 28, 29, 49, 89"
* Application exits with error code (non-zero)
* No database connection attempted
* No output files created
* Error logged to error log file

**Test Data:**

* Invalid SC codes to test: 1, 5, 10, 15, 25, 50, 99, 100

**Pass/Fail Criteria:**

* ✓ PASS: Error message displayed, clean exit, no files created
* ✗ FAIL: Application processes invalid SC or crashes

**TC\_SIA\_N\_002: Invalid UPDTFLAG Value**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_002 |
| **Test Case Name** | Invalid UPDTFLAG Rejection |
| **Test Type** | Functional - Negative |
| **Priority** | High |
| **Module** | Input Validation |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 X
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Yes
3. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 1
4. Observe errors for each

**Expected Results:**

* Valid UPDTFLAG values: Y, y, N, n only
* Invalid values rejected:
  + X: Error "Invalid UPDTFLAG: X. Must be Y/y/N/n"
  + Yes: Error "Invalid UPDTFLAG: Yes. Must be Y/y/N/n"
  + 1: Error "Invalid UPDTFLAG: 1. Must be Y/y/N/n"
* Application exits without processing
* No database connection

**Pass/Fail Criteria:**

* ✓ PASS: All invalid values rejected with clear error messages
* ✗ FAIL: Invalid value accepted or unclear error

**TC\_SIA\_N\_003: Missing Required Arguments**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_003 |
| **Test Case Name** | Missing Command Line Arguments |
| **Test Type** | Functional - Negative |
| **Priority** | High |
| **Module** | Input Validation |

**Test Steps:**

1. Execute: java -jar sia-extract.jar (no arguments)
2. Execute: java -jar sia-extract.jar -fix (missing SC, LOADDT, UPDTFLAG)
3. Execute: java -jar sia-extract.jar -fix 07 (missing LOADDT, UPDTFLAG)
4. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 (missing UPDTFLAG)

**Expected Results:**

* Usage message displayed for all cases:
* Usage: java -jar sia-extract.jar [-version] | [-fix SC[0=all] LOADDT UPDTFLAG[Y/y/N/n]]Options: -version Display version information -fix SC LOADDT FLAG Extract records for service centerArguments: SC Service center code (7,8,9,17,18,19,28,29,49,89) or 0 for all LOADDT Load date in MM/DD/YYYY format UPDTFLAG Y/y to update database, N/n for read-onlyExamples: java -jar sia-extract.jar -version java -jar sia-extract.jar -fix 07 09/29/2025 Y java -jar sia-extract.jar -fix 0 09/29/2025 N
* Application exits with error code
* No processing attempted

**Pass/Fail Criteria:**

* ✓ PASS: Clear usage message, no processing, clean exit
* ✗ FAIL: Confusing error or attempted processing

**TC\_SIA\_N\_004: Database Connection Failure**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_004 |
| **Test Case Name** | Handle Database Connection Failure |
| **Test Type** | Functional - Negative |
| **Priority** | Critical |
| **Module** | Database Connection |

**Prerequisites:**

* Stop database service or modify connection properties to invalid values

**Test Steps:**

1. Stop Oracle database service
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Observe error handling
4. Restart database

**Expected Results:**

* Connection attempt fails
* Error message displayed: "UNABLE TO CONNECT TO DATABASE: [Oracle Error Details]"
* SQLException details logged to error log
* Error code: ORA-12541 (listener not started) or similar
* Application exits gracefully (no crash)
* Exit code: non-zero (error)
* Partial output files not created

**Pass/Fail Criteria:**

* ✓ PASS: Clear error message, graceful exit, error logged
* ✗ FAIL: Application hangs, crashes, or unclear error

**TC\_SIA\_N\_005: Invalid Credentials**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_005 |
| **Test Case Name** | Handle Invalid Database Credentials |
| **Test Type** | Security - Negative |
| **Priority** | Critical |
| **Module** | Authentication |

**Prerequisites:**

* Modify configuration file with incorrect password

**Test Steps:**

1. Edit sia-config.properties: change database password to incorrect value
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Observe error
4. Restore correct password

**Expected Results:**

* Authentication fails
* Error: "UNABLE TO CONNECT TO DATABASE: Invalid username/password"
* SQLException: ORA-01017 (invalid username/password)
* Error logged without exposing password in logs
* No processing attempted
* Application exits with error code

**Pass/Fail Criteria:**

* ✓ PASS: Authentication rejected, secure error handling, password not in logs
* ✗ FAIL: Connection succeeds with wrong password or password exposed in logs

**TC\_SIA\_N\_006: No Records Found for Date**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_006 |
| **Test Case Name** | Handle No Matching Records Scenario |
| **Test Type** | Functional - Negative |
| **Priority** | Medium |
| **Module** | Data Extraction |

**Prerequisites:**

* Use LOADDT with no matching records (e.g., '12/31/2099')

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 07 12/31/2099 Y
2. Observe console output
3. Check for output file creation

**Expected Results:**

* Database connection successful
* Query executes successfully (no SQL error)
* Zero records returned
* Console message: "No records found for SC 07 - LOADDT 12/31/2099"
* No output file created (or existing file unchanged if append mode)
* Application exits with success code (0) - not an error condition
* No database updates performed

**Pass/Fail Criteria:**

* ✓ PASS: Appropriate message, no file created, clean exit
* ✗ FAIL: Error thrown, empty file created, or application crash

**TC\_SIA\_N\_007: File Write Permission Denied**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_007 |
| **Test Case Name** | Handle Output Directory Write Permission Failure |
| **Test Type** | Functional - Negative |
| **Priority** | High |
| **Module** | File I/O |

**Prerequisites:**

* Remove write permissions from SC07/ directory

**Test Steps:**

1. Execute: chmod 555 SC07/ (Linux/Unix)
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Observe error handling
4. Restore permissions: chmod 755 SC07/

**Expected Results:**

* Records extracted from database
* File write attempt fails
* Error: "Unable to write to output file: SC07/alsent.07.20250929 - Permission denied"
* IOException caught and handled
* Error logged to error log
* Application exits with error code
* **Database NOT updated** (transaction rolled back if updtflag=Y)

**Pass/Fail Criteria:**

* ✓ PASS: Clear error, no database update, graceful exit
* ✗ FAIL: Application crash, database updated without file, or unclear error

**TC\_SIA\_N\_008: Invalid Date Format**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_008 |
| **Test Case Name** | Handle Invalid LOADDT Format |
| **Test Type** | Functional - Negative |
| **Priority** | Medium |
| **Module** | Input Validation |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 07 2025-09-29 Y (ISO format)
2. Execute: java -jar sia-extract.jar -fix 07 09/32/2025 Y (invalid day)
3. Execute: java -jar sia-extract.jar -fix 07 ABCD Y (non-date)

**Expected Results:**

* Application validates date format before database query
* Error: "Invalid date format: [value]. Expected format: MM/DD/YYYY"
* **For '2025-09-29'**: "Invalid date format: 2025-09-29. Expected format: MM/DD/YYYY"
* **For '09/32/2025'**: "Invalid date: 09/32/2025. Day must be 1-31"
* **For 'ABCD'**: "Invalid date format: ABCD. Expected format: MM/DD/YYYY"
* Application exits without database connection
* No processing attempted

**Pass/Fail Criteria:**

* ✓ PASS: Date validation catches errors before database query
* ✗ FAIL: Invalid dates passed to database or application crash

**TC\_SIA\_N\_009: SQL Query Failure**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_009 |
| **Test Case Name** | Handle SQL Query Execution Failure |
| **Test Type** | Functional - Negative |
| **Priority** | Medium |
| **Module** | Database Query |

**Prerequisites:**

* Temporarily revoke SELECT privilege on DIAL.COREDIAL table

**Test Steps:**

1. As DBA: REVOKE SELECT ON DIAL.COREDIAL FROM test\_als\_user;
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Observe error
4. Restore privilege: GRANT SELECT ON DIAL.COREDIAL TO test\_als\_user;

**Expected Results:**

* Initial connection succeeds
* TIN hook query fails with SQLException
* Error: "UNABLE TO GET TINHOOKS: [SQL Error Details]"
* ORA-00942: table or view does not exist (or ORA-01031: insufficient privileges)
* Full stack trace logged to error log
* User-friendly error message on console
* Application exits with error code
* No partial processing

**Pass/Fail Criteria:**

* ✓ PASS: SQL error caught, logged, clear user message, clean exit
* ✗ FAIL: Application crash, unclear error, or security details exposed

**TC\_SIA\_N\_010: Database Update Failure - Table Locked**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_N\_010 |
| **Test Case Name** | Handle Database Update Failure Due to Lock |
| **Test Type** | Functional - Negative |
| **Priority** | High |
| **Module** | Database Update |

**Prerequisites:**

* Second database session to lock table
* UPDTFLAG = Y

**Test Steps:**

1. In Session 2: LOCK TABLE ALS.ASSN IN EXCLUSIVE MODE;
2. In Session 1: Execute java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Observe behavior (wait timeout or error)
4. In Session 2: ROLLBACK; (release lock)

**Expected Results:**

* File extraction completes successfully
* UPDATE statement attempts execution
* Blocks waiting for lock release OR
* Times out after configured timeout (e.g., 30 seconds)
* Error: "PROBLEM UPDATE 1: Unable to update database - resource busy"
* SQLException: ORA-00054 (resource busy and acquire with NOWAIT specified)
* Transaction rolled back
* No partial updates
* Error logged
* Application exits with error code

**Pass/Fail Criteria:**

* ✓ PASS: Timeout handled, transaction rolled back, error logged
* ✗ FAIL: Application hangs indefinitely or partial updates committed

**6. Boundary Value Analysis (BVA)**

**TC\_SIA\_BVA\_001: Service Center Code - Minimum Valid**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_001 |
| **Test Case Name** | SC Code Minimum Boundary |
| **Test Type** | BVA - Valid Minimum |
| **Priority** | High |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 7 09/29/2025 Y

**Expected Results:**

* SC 7 (minimum valid code) accepted
* Processing completes successfully
* Output file: SC07/alsent.07.20250929

**Test Data:** SC = 7

**Pass/Fail:** ✓ SC 7 processed correctly

**TC\_SIA\_BVA\_002: Service Center Code - Maximum Valid**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_002 |
| **Test Case Name** | SC Code Maximum Boundary |
| **Test Type** | BVA - Valid Maximum |
| **Priority** | High |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 89 09/29/2025 Y

**Expected Results:**

* SC 89 (maximum valid code) accepted
* Processing completes successfully
* Output file: SC89/alsent.89.20250929

**Test Data:** SC = 89

**Pass/Fail:** ✓ SC 89 processed correctly

**TC\_SIA\_BVA\_003: Service Center Code - Just Below Minimum**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_003 |
| **Test Case Name** | SC Code Below Minimum Boundary |
| **Test Type** | BVA - Invalid Below Min |
| **Priority** | High |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 6 09/29/2025 Y

**Expected Results:**

* SC 6 rejected
* Error: "Invalid SC Code: 6"

**Pass/Fail:** ✓ SC 6 correctly rejected

**TC\_SIA\_BVA\_004: Service Center Code - Just Above Maximum**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_004 |
| **Test Case Name** | SC Code Above Maximum Boundary |
| **Test Type** | BVA - Invalid Above Max |
| **Priority** | High |

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 90 09/29/2025 Y

**Expected Results:**

* SC 90 rejected
* Error: "Invalid SC Code: 90"

**Pass/Fail:** ✓ SC 90 correctly rejected

**TC\_SIA\_BVA\_005: TIN Value - Minimum (1 digit)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_005 |
| **Test Case Name** | TIN Minimum Value Formatting |
| **Test Type** | BVA - Valid Minimum |
| **Priority** | Medium |

**Prerequisites:**

* Test TIN = 1 in database

**Test Steps:**

1. Execute extraction
2. Verify TIN formatting in output

**Expected Results:**

* TIN = 1 formatted as 9 digits: 000000001
* Output record contains: 07620000000001...
* Left-padded with zeros

**Pass/Fail:** ✓ TIN 1 formatted correctly

**TC\_SIA\_BVA\_006: TIN Value - Maximum (9 digits)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_006 |
| **Test Case Name** | TIN Maximum Value Formatting |
| **Test Type** | BVA - Valid Maximum |
| **Priority** | Medium |

**Prerequisites:**

* Test TIN = 999999999

**Test Steps:**

1. Execute extraction
2. Verify TIN in output

**Expected Results:**

* TIN = 999999999 formatted correctly: 999999999
* Output record contains: 07620999999999...
* No overflow

**Pass/Fail:** ✓ TIN 999999999 formatted correctly

**TC\_SIA\_BVA\_007: TIN Array Capacity - At Limit (30,000 records)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_007 |
| **Test Case Name** | TIN Array Maximum Capacity |
| **Test Type** | BVA - Performance/Scalability |
| **Priority** | Critical |

**Prerequisites:**

* Populate database with exactly 30,000 TINs for test date

**Test Steps:**

1. Create 30,000 test TIN records
2. Execute: java -jar sia-extract.jar -fix 0 {test\_date} Y
3. Monitor processing
4. Verify all 30,000 records processed

**Expected Results:**

* All 30,000 TINs retrieved
* All processed successfully
* No array overflow errors
* All records written to output files
* Processing completes (may take several minutes)
* Database updates successful for all records

**Test Data:** 30,000 TINs

**Pass/Fail Criteria:**

* ✓ PASS: All 30,000 records processed
* ✗ FAIL: Array overflow, OutOfMemoryError, or incomplete processing

**Notes:** Original legacy code had hard limit of 30,000. Verify Java implementation handles this appropriately (dynamic arrays or pagination).

**TC\_SIA\_BVA\_008: TIN Array Capacity - Exceeding Limit (35,000 records)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_008 |
| **Test Case Name** | TIN Array Beyond Capacity |
| **Test Type** | BVA - Negative (Overflow) |
| **Priority** | Critical |

**Prerequisites:**

* Populate database with 35,000 TINs

**Test Steps:**

1. Create 35,000 test TIN records
2. Execute extraction
3. Monitor for errors

**Expected Results - Java Implementation Should:**

* **Option A (Dynamic)**: Process all 35,000 records using ArrayList or pagination
* **Option B (Fixed Array)**: Display error: "Record count (35000) exceeds maximum capacity (30000)"
* **Preferred**: Dynamic handling to process all records
* No memory corruption or crash
* Clear error message if limit enforced

**Pass/Fail Criteria:**

* ✓ PASS: All records processed OR clear capacity error with safe exit
* ✗ FAIL: Crash, memory corruption, or silent data loss

**TC\_SIA\_BVA\_009: Date - Minimum Valid (01/01/1900)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_009 |
| **Test Case Name** | NULLDATE Processing |
| **Test Type** | BVA - Special Value |
| **Priority** | High |

**Prerequisites:**

* Records exist with LOADDT = '01/01/1900' (NULLDATE)

**Test Steps:**

1. Execute: java -jar sia-extract.jar -fix 07 01/01/1900 Y
2. Verify extraction
3. Check database updates

**Expected Results:**

* Records with LOADDT = '01/01/1900' retrieved
* Processed normally
* If UPDTFLAG=Y: LOADDT updated to system date
* Special handling for NULLDATE (bulk update at end)

**Pass/Fail:** ✓ NULLDATE records processed correctly

**TC\_SIA\_BVA\_010: Financial System Code - All Valid Values**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_010 |
| **Test Case Name** | All Valid FS Codes Processing |
| **Test Type** | BVA - Valid Partitions |
| **Priority** | High |

**Prerequisites:**

* Test records with fs = 1, 2, 3, 4, 6

**Test Steps:**

1. Execute extraction
2. Verify each FS code processes correctly
3. Check output format for each

**Expected Results:**

* **fs = 1 (IMF)**: IFEUP format (if trans\_mod=2 or 3)
* **fs = 2 (BMF)**: BFEUP format
* **fs = 3 (EPEF)**: BFEUP format
* **fs = 4 (IRAF)**: IFEUP format
* **fs = 6 (NMF)**: BFEUP format **with ULC code**

**Pass/Fail:** ✓ All FS codes processed with correct formats

**TC\_SIA\_BVA\_011: Financial System Code - Invalid Value (fs=5)**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_BVA\_011 |
| **Test Case Name** | Invalid FS Code Handling |
| **Test Type** | BVA - Invalid Value |
| **Priority** | Medium |

**Prerequisites:**

* Manually insert record with fs = 5 (not in valid set)

**Test Steps:**

1. Create test TIN with fs = 5
2. Execute extraction
3. Observe handling

**Expected Results:**

* **Option A**: Record skipped with warning: "Invalid FS code (5) for TIN [xxxxx] - skipping"
* **Option B**: Default BFEUP format applied
* No application crash
* Error logged

**Pass/Fail:** ✓ Invalid FS handled gracefully (no crash)

**7. Integration Test Cases**

**TC\_SIA\_INT\_001: End-to-End Processing - Single SC**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_INT\_001 |
| **Test Case Name** | Complete End-to-End Workflow Verification |
| **Test Type** | Integration |
| **Priority** | Critical |

**Test Steps:**

1. Prepare test dataset: 100 TINs for SC 07, mixed trans\_mod values
2. Execute: java -jar sia-extract.jar -fix 07 09/29/2025 Y
3. Verify all stages

**Expected Results:**

1. **Initialization**: Application starts, config loaded
2. **Validation**: Arguments validated successfully
3. **Database Connection**: Connection established
4. **TIN Retrieval**: All 100 TINs retrieved
5. **Detail Processing**: Each TIN queried for details
6. **Record Formatting**: Records formatted per trans\_mod
7. **File Output**: Output file created with all eligible records
8. **Database Update**: LOADDT and QPICKIND updated
9. **Cleanup**: Database commit, file closed, clean exit

**Pass/Fail:** ✓ All stages complete successfully

**TC\_SIA\_INT\_002: Multi-SC Processing**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_INT\_002 |
| **Test Case Name** | Multiple Service Center Processing |
| **Test Type** | Integration |
| **Priority** | High |

**Test Steps:**

1. Prepare data for SC 7, 8, 9, 17 (50 TINs each)
2. Execute: java -jar sia-extract.jar -fix 0 09/29/2025 Y
3. Verify files for all SCs

**Expected Results:**

* 4 output files created
* Each SC's records properly segregated
* No cross-contamination (SC 7 TINs only in SC07 file)
* All database updates correct

**Pass/Fail:** ✓ All SC files correct and segregated

**TC\_SIA\_INT\_003: Database Transaction Integrity**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_INT\_003 |
| **Test Case Name** | Database Transaction Commit/Rollback |
| **Test Type** | Integration - Data Integrity |
| **Priority** | Critical |

**Test Steps:**

1. Execute extraction with UPDTFLAG=Y
2. Simulate mid-process failure (kill application at 50% completion)
3. Query database for partial updates
4. Re-run extraction

**Expected Results:**

* **If transaction-based**: All updates rolled back on failure (ACID compliance)
* No partial updates (all or nothing)
* Re-run processes all records including previously "completed" ones
* **If record-by-record commit**: Some records updated, re-run handles correctly

**Pass/Fail:** ✓ Transaction integrity maintained

**8. Performance Test Cases**

**TC\_SIA\_PERF\_001: Large Dataset Performance**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_PERF\_001 |
| **Test Case Name** | Performance with Large Record Set |
| **Test Type** | Performance |
| **Priority** | High |

**Test Steps:**

1. Load 10,000 TINs into database
2. Execute extraction
3. Measure execution time and resource usage

**Expected Results:**

* Processing completes within acceptable timeframe (e.g., < 5 minutes for 10,000 records)
* Memory usage stable (no memory leaks)
* CPU usage reasonable
* Database connections properly managed (no connection leaks)

**Performance Benchmarks:**

* 10,000 records: < 5 minutes
* 20,000 records: < 10 minutes
* 30,000 records: < 15 minutes

**Pass/Fail:** ✓ Performance within acceptable limits

**TC\_SIA\_PERF\_002: Memory Consumption**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_PERF\_002 |
| **Test Case Name** | Memory Usage Monitoring |
| **Test Type** | Performance |
| **Priority** | Medium |

**Test Steps:**

1. Monitor JVM memory before execution
2. Execute with 30,000 TINs
3. Monitor heap usage during processing
4. Verify memory released after completion

**Expected Results:**

* Initial heap: ~50MB
* Peak heap: < 512MB for 30,000 records
* Memory released after processing
* No OutOfMemoryError
* Garbage collection operates normally

**Pass/Fail:** ✓ Memory usage within limits, no leaks

**TC\_SIA\_PERF\_003: Database Query Optimization**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_PERF\_003 |
| **Test Case Name** | Query Performance Analysis |
| **Test Type** | Performance |
| **Priority** | Medium |

**Test Steps:**

1. Enable SQL tracing
2. Execute extraction
3. Review query execution plans
4. Identify slow queries

**Expected Results:**

* TIN hook query uses appropriate indexes
* Detail queries execute quickly (< 100ms each)
* No full table scans on large tables
* UPDATE statements use ROWID (fastest access)

**Pass/Fail:** ✓ Queries optimized with proper indexes

**9. Security Test Cases**

**TC\_SIA\_SEC\_001: Credential Protection**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_SEC\_001 |
| **Test Case Name** | Database Credential Security |
| **Test Type** | Security |
| **Priority** | Critical |

**Test Steps:**

1. Review configuration file
2. Check for plaintext passwords
3. Review application logs
4. Check error messages

**Expected Results:**

* Passwords encrypted in configuration
* No passwords in logs (even debug logs)
* Error messages don't expose credentials
* Connection strings don't contain passwords

**Pass/Fail:** ✓ No credential exposure

**TC\_SIA\_SEC\_002: SQL Injection Prevention**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_SEC\_002 |
| **Test Case Name** | SQL Injection Attack Prevention |
| **Test Type** | Security |
| **Priority** | High |

**Test Steps:**

1. Attempt SQL injection in LOADDT: ' OR '1'='1
2. Attempt injection in SC parameter
3. Verify parameterized queries used

**Expected Results:**

* All user inputs validated before database use
* Prepared statements/parameterized queries used
* No dynamic SQL string concatenation
* Injection attempts rejected or sanitized

**Pass/Fail:** ✓ SQL injection prevented

**TC\_SIA\_SEC\_003: File Permission Security**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_SEC\_003 |
| **Test Case Name** | Output File Security Permissions |
| **Test Type** | Security |
| **Priority** | Medium |

**Test Steps:**

1. Execute extraction
2. Check output file permissions (Unix: ls -l)
3. Verify appropriate access restrictions

**Expected Results:**

* Output files: 640 (rw-r-----) or more restrictive
* Owned by executing user
* Not world-readable
* Contains sensitive taxpayer data - must be protected

**Pass/Fail:** ✓ File permissions appropriately restrictive

**10. Data Integrity Test Cases**

**TC\_SIA\_DATA\_001: Record Format Accuracy**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_DATA\_001 |
| **Test Case Name** | Output Record Format Validation |
| **Test Type** | Data Integrity |
| **Priority** | Critical |

**Test Steps:**

1. Extract records
2. Parse output file programmatically
3. Verify field positions and values

**Expected Results:**

* All records exactly 55 characters (including newline)
* Fixed-width fields properly aligned
* No extra spaces or characters
* All required fields populated
* Numeric fields properly formatted

**Pass/Fail:** ✓ All records formatted correctly

**TC\_SIA\_DATA\_002: Database Update Accuracy**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_DATA\_002 |
| **Test Case Name** | Database Update Value Verification |
| **Test Type** | Data Integrity |
| **Priority** | Critical |

**Test Steps:**

1. Document original values for test TINs
2. Execute with UPDTFLAG=Y
3. Query updated values
4. Compare expected vs actual

**Expected Results:**

* LOADDT updated to exact system date (MM/DD/YYYY format)
* QPICKIND updated from 3 to 4
* TRANSMOD updated to correct trans\_mod value
* Only processed records updated
* Non-eligible records (trans\_mod=0) not updated

**Pass/Fail:** ✓ All updates accurate

**TC\_SIA\_DATA\_003: Data Consistency - File vs Database**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_DATA\_003 |
| **Test Case Name** | File Output and Database Consistency |
| **Test Type** | Data Integrity |
| **Priority** | High |

**Test Steps:**

1. Execute extraction
2. Parse TINs from output file
3. Query database for same TINs
4. Compare values (FS, TT, nmctrl, aoto, ronum)

**Expected Results:**

* Every TIN in output file exists in database
* All field values match between file and database
* No discrepancies in data
* Record counts match

**Pass/Fail:** ✓ File and database data consistent

**TC\_SIA\_DATA\_004: Character Encoding Validation**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC\_SIA\_DATA\_004 |
| **Test Case Name** | Special Character and Encoding Handling |
| **Test Type** | Data Integrity |
| **Priority** | Medium |

**Test Steps:**

1. Insert test records with special characters in nmctrl field
2. Execute extraction
3. Verify character preservation

**Expected Results:**

* ASCII characters preserved correctly
* No character corruption
* Output file readable with standard text editors
* UTF-8 or ASCII encoding consistent

**Pass/Fail:** ✓ Characters preserved correctly

**11. Traceability Matrix**

| **Requirement** | **Test Case(s)** | **Priority** | **Status** |
| --- | --- | --- | --- |
| Extract records for single SC | TC\_SIA\_F\_002 | High |  |
| Extract records for all SCs | TC\_SIA\_F\_003 | High |  |
| Update database when UPDTFLAG=Y | TC\_SIA\_F\_004 | Critical |  |
| Read-only when UPDTFLAG=N | TC\_SIA\_F\_005 | Critical |  |
| Generate TSIGN records | TC\_SIA\_F\_006 | High |  |
| Generate TC015 records | TC\_SIA\_F\_007, TC\_SIA\_F\_008 | High |  |
| Generate TC030 records | TC\_SIA\_F\_009 | High |  |
| Skip non-eligible records | TC\_SIA\_F\_010 | High |  |
| Validate SC codes | TC\_SIA\_N\_001, TC\_SIA\_BVA\_001-004 | High |  |
| Handle invalid inputs | TC\_SIA\_N\_002, TC\_SIA\_N\_003, TC\_SIA\_N\_008 | High |  |
| Handle database errors | TC\_SIA\_N\_004, TC\_SIA\_N\_005, TC\_SIA\_N\_009, TC\_SIA\_N\_010 | Critical |  |
| Handle file I/O errors | TC\_SIA\_N\_007 | High |  |
| Process large datasets | TC\_SIA\_BVA\_007, TC\_SIA\_BVA\_008, TC\_SIA\_PERF\_001 | Critical |  |
| Maintain data integrity | TC\_SIA\_DATA\_001-004 | Critical |  |
| Protect credentials | TC\_SIA\_SEC\_001 | Critical |  |
| Prevent SQL injection | TC\_SIA\_SEC\_002 | High |  |
| Include ULC for NMF only | TC\_SIA\_F\_008, TC\_SIA\_BVA\_010 | High |  |
| Append posting delay code | TC\_SIA\_F\_007, TC\_SIA\_F\_009 | High |  |

**12. Test Execution Summary Template**

| **Test Case ID** | **Test Case Name** | **Executed By** | **Date** | **Result** | **Defects** | **Comments** |
| --- | --- | --- | --- | --- | --- | --- |
| TC\_SIA\_F\_001 | Display Version |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_002 | Single SC Extract |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_003 | All SC Extract |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_004 | Update with Y Flag |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_005 | Read-Only with N Flag |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_006 | TSIGN Format |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_007 | TC015 IFEUP Format |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_008 | TC015 BFEUP NMF |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_009 | TC030 Format |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_F\_010 | Skip Non-Eligible |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_001 | Invalid SC |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_002 | Invalid UPDTFLAG |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_003 | Missing Arguments |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_004 | DB Connection Fail |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_005 | Invalid Credentials |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_006 | No Records Found |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_007 | Write Permission Fail |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_008 | Invalid Date Format |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_009 | SQL Query Failure |  |  | ☐ Pass ☐ Fail |  |  |
| TC\_SIA\_N\_010 | Table Locked |  |  | ☐ Pass ☐ Fail |  |  |

**Appendix A: Test Data Setup Scripts**

-- Sample test data creation script

-- Create test TINs for various scenarios

-- TSIGN eligible records (trans\_mod=1)

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000001, 1, 4, '09/29/2025', 'O', 3, 5, 7, 12, 3,

'AB12', NULL, SYSDATE, 1, 1);

-- TC015 eligible records (trans\_mod=2, IMF)

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000002, 1, 4, '09/29/2025', 'O', 3, 8, 9, 15, 6,

'XY34', NULL, SYSDATE, 1, 2);

-- TC015 eligible records (trans\_mod=2, NMF with ULC)

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000003, 6, 4, '09/29/2025', 'O', 3, 10, 11, 20, 8,

'CD56', 25, SYSDATE, 1, 2);

-- TC030 eligible records (trans\_mod=3, BMF)

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000004, 2, 4, '09/29/2025', 'O', 3, 12, 13, 25, 9,

'EF78', NULL, SYSDATE, 1, 3);

-- Not eligible records (trans\_mod=0)

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000005, 1, 4, '09/29/2025', 'O', 3, 5, 7, 12, 3,

'GH90', NULL, SYSDATE, 1, 0);

-- NULLDATE records

INSERT INTO ALS.ASSN

(TIN, FS, TT, LOADDT, STATUS, QPICKIND, AREA, TERRITORY, GRNUM, RONUM,

NMCTRL, ULC\_CD, SEL\_DATE, ROMNUM, CORESID)

VALUES

(100000006, 1, 4, '01/01/1900', 'O', 3, 5, 7, 12, 3,

'IJ12', NULL, SYSDATE, 1, 1);

-- Insert corresponding COREDIAL records

INSERT INTO DIAL.COREDIAL (TIN, FS, TT, CORETIN, COREFS, CORETT)

SELECT TIN, FS, TT, TIN, FS, TT

FROM ALS.ASSN

WHERE TIN BETWEEN 100000001 AND 100000006;

-- Insert DOMAP entries for SC mapping

INSERT INTO DOMAP (SC, AREA) VALUES (7, 5);

INSERT INTO DOMAP (SC, AREA) VALUES (7, 8);

INSERT INTO DOMAP (SC, AREA) VALUES (7, 10);

INSERT INTO DOMAP (SC, AREA) VALUES (7, 12);

COMMIT;

**Appendix B: Sample Output Format Reference**

**TSIGN Record Format**

Position Length Field Example Description

-------- ------ -------------- ---------- ---------------------------

1-2 2 SC Code 07 Service Center

3-5 3 Constant 620 Fixed value

6-14 9 TIN 100000001 Tax Identification Number

15 1 FS 1 Financial System

16 1 TT 4 Transaction Type

17-28 12 Zeros 000...000 Fixed padding

29-30 2 Constant U0 Fixed value

31-35 5 Trans Type TSIGN Transaction indicator

36-39 4 NMCTRL AB12 Name Control

40-43 4 RONUM 3456 Routing Number

44-47 4 AOTO 7890 Account Origin Territory

48-53 6 Padding (spaces) Fixed spacing

54 1 Constant 0 Fixed value

55 1 Newline \n Line terminator

**TC015 IFEUP Record Format (IMF/IRAF)**

Position Length Field Example Description

-------- ------ -------------- ---------- ---------------------------

1-2 2 SC Code 07 Service Center

3-5 3 Constant 620 Fixed value

6-14 9 TIN 100000002 Tax Identification Number

15 1 FS 1 Financial System

16 1 TT 4 Transaction Type

17-28 12 Zeros 000...000 Fixed padding

29-30 2 Constant U0 Fixed value

31-39 9 Trans Type IFEUP015 Transaction indicator

40-43 4 NMCTRL XY34 Name Control

44-46 3 Constant 500 Fixed value

47-48 2 CLC 12 First 2 chars of AOTO

49-50 2 Period/Const .2 Fixed value

51-52 2 Constant s% Fixed value

53-54 2 Constant 4s Fixed value

55-56 2 Constant %4 Fixed value

57-58 2 Constant s1 Fixed value

59 1 Posting Delay 1 Delay code

60 1 Newline \n Line terminator

**TC015 BFEUP Record Format (NMF with ULC)**

Position Length Field Example Description

-------- ------ -------------- ---------- ---------------------------

31-39 9 Trans Type BFEUP015 Transaction indicator

40-43 4 NMCTRL CD56 Name Control

44-45 2 Constant 50 Fixed value (not 500)

46-47 2 ULC Code 25 Unit Locator Code

48-49 2 CLC 90 First 2 chars of AOTO

(remaining fields similar to IFEUP)

59 1 Posting Delay 1 Delay code

60 1 Newline \n Line terminator

**Appendix C: Error Code Reference**

| **Error Code** | **Error Message** | **Cause** | **Resolution** |
| --- | --- | --- | --- |
| ERR-001 | Invalid SC Code | SC code not in approved list | Use valid SC: 7,8,9,17,18,19,28,29,49,89 |
| ERR-002 | Invalid UPDTFLAG | UPDTFLAG not Y/y/N/n | Use Y, y, N, or n only |
| ERR-003 | Missing Arguments | Required arguments not provided | Provide all required parameters |
| ERR-004 | Invalid Date Format | LOADDT not in MM/DD/YYYY format | Use MM/DD/YYYY format |
| ERR-005 | Database Connection Failed | Cannot connect to database | Check database status and credentials |
| ERR-006 | Authentication Failed | Invalid username/password | Verify credentials in config file |
| ERR-007 | SQL Query Failed | Query execution error | Check table permissions and syntax |
| ERR-008 | File Write Error | Cannot write to output file | Check directory permissions |
| ERR-009 | Database Update Failed | UPDATE statement failed | Check table lock status |
| ERR-010 | Configuration Error | Config file missing/invalid | Verify sia-config.properties exists |

**Appendix D: Configuration File Template**

**sia-config.properties**

# Database Configuration

db.url=jdbc:oracle:thin:@hostname:1521:SID

db.username=als

db.password.encrypted=<encrypted\_password>

db.schema.als=ALS

db.schema.dial=DIAL

# Connection Pool Settings

db.pool.minSize=2

db.pool.maxSize=10

db.pool.timeout=30

# File Output Settings

output.directory=/data/sia-extract

output.filePermissions=640

output.createDirectories=true

# Application Settings

app.version=2.0

app.maxRecords=50000

app.batchSize=1000

# Logging Configuration

log.directory=/var/log/sia-extract

log.level=INFO

log.maxFileSize=10MB

log.maxBackups=5

# Error Handling

error.log.enabled=true

error.log.path=/var/log/sia-extract/error.log

error.emailNotification=false

# Performance Tuning

query.fetchSize=500

query.timeout=300

# Valid Service Center Codes

valid.sc.codes=7,8,9,17,18,19,28,29,49,89

**Appendix E: Test Environment Checklist**

**Pre-Test Verification**

* [ ] Oracle database running and accessible
* [ ] Test schemas (ALS, DIAL) created
* [ ] Required tables exist: ALS.ASSN, DIAL.COREDIAL, DOMAP
* [ ] Test user has appropriate privileges:
  + [ ] SELECT on ALS.ASSN
  + [ ] SELECT on DIAL.COREDIAL
  + [ ] SELECT on DOMAP
  + [ ] UPDATE on ALS.ASSN
* [ ] Java JRE 8+ installed
* [ ] Application JAR file deployed
* [ ] Configuration file properly configured
* [ ] Output directories created with write permissions:
  + [ ] SC07/
  + [ ] SC08/
  + [ ] SC09/
  + [ ] SC17/
  + [ ] SC18/
  + [ ] SC19/
  + [ ] SC28/
  + [ ] SC29/
  + [ ] SC49/
  + [ ] SC89/
* [ ] Error log directory exists with write permissions
* [ ] Test data loaded (see Appendix A)
* [ ] Database backup completed (for rollback if needed)

**Post-Test Verification**

* [ ] All test cases executed and results documented
* [ ] Defects logged in defect tracking system
* [ ] Test data cleaned up or database restored
* [ ] Output files archived for reference
* [ ] Test execution summary completed
* [ ] Sign-off obtained from stakeholders

**Appendix F: Defect Report Template**

**Defect Report Form**

**Defect ID:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Date Reported:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Reported By:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Test Case ID:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Test Case Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Severity:**  
☐ Critical - Application crash, data loss, security vulnerability  
☐ High - Major functionality broken, no workaround  
☐ Medium - Feature not working as expected, workaround available  
☐ Low - Minor issue, cosmetic, enhancement

**Priority:**  
☐ P1 - Must fix before release  
☐ P2 - Should fix before release  
☐ P3 - Fix in future release  
☐ P4 - Enhancement/Nice to have

**Environment:**

* OS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Java Version: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Database Version: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Application Version: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Steps to Reproduce:**

**Expected Result:**

**Actual Result:**

**Screenshots/Logs:** (Attach)

**Additional Notes:**

**Assigned To:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Status:** ☐ New ☐ In Progress ☐ Fixed ☐ Verified ☐ Closed

**Appendix G: Performance Benchmarks**

**Expected Performance Metrics**

| **Metric** | **Target** | **Acceptable** | **Unacceptable** |
| --- | --- | --- | --- |
| 1,000 records | < 30 sec | < 60 sec | > 60 sec |
| 5,000 records | < 2 min | < 5 min | > 5 min |
| 10,000 records | < 5 min | < 10 min | > 10 min |
| 20,000 records | < 10 min | < 20 min | > 20 min |
| 30,000 records | < 15 min | < 30 min | > 30 min |

**Resource Utilization Targets**

| **Resource** | **Target** | **Maximum** |
| --- | --- | --- |
| Memory (Heap) | < 256 MB | < 512 MB |
| CPU Usage | < 50% | < 80% |
| Database Connections | 2-5 | 10 |
| File I/O Wait | < 10% | < 20% |

**Appendix H: Regression Test Suite**

**Critical Path Tests (Execute for Every Build)**

1. TC\_SIA\_F\_002 - Single SC extraction
2. TC\_SIA\_F\_004 - Database update with Y flag
3. TC\_SIA\_F\_005 - Read-only with N flag
4. TC\_SIA\_F\_006 - TSIGN format
5. TC\_SIA\_N\_001 - Invalid SC code rejection
6. TC\_SIA\_N\_004 - Database connection failure
7. TC\_SIA\_DATA\_002 - Database update accuracy

**Full Regression Suite (Execute Before Release)**

* All functional test cases (TC\_SIA\_F\_001 through TC\_SIA\_F\_010)
* All negative test cases (TC\_SIA\_N\_001 through TC\_SIA\_N\_010)
* Critical BVA tests (TC\_SIA\_BVA\_007, TC\_SIA\_BVA\_008)
* Integration tests (TC\_SIA\_INT\_001 through TC\_SIA\_INT\_003)
* Data integrity tests (TC\_SIA\_DATA\_001 through TC\_SIA\_DATA\_004)
* Security tests (TC\_SIA\_SEC\_001 through TC\_SIA\_SEC\_003)

**Appendix I: Sign-Off Sheet**

**Test Completion Sign-Off**

**Project:** SIA Extract Java Implementation  
**Test Phase:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Unit/Integration/System/UAT)  
**Test Period:** From \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ To \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Test Metrics**

| **Metric** | **Count** |
| --- | --- |
| Total Test Cases |  |
| Test Cases Executed |  |
| Test Cases Passed |  |
| Test Cases Failed |  |
| Test Cases Blocked |  |
| Test Cases Not Run |  |
| **Pass Rate** | **\_\_\_\_%** |

**Defect Summary**

| **Severity** | **Total** | **Open** | **Fixed** | **Verified** |
| --- | --- | --- | --- | --- |
| Critical |  |  |  |  |
| High |  |  |  |  |
| Medium |  |  |  |  |
| Low |  |  |  |  |
| **Total** |  |  |  |  |

**Outstanding Issues**

List any unresolved critical or high priority defects:

**Approval Signatures**

**QA Lead:**  
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Development Lead:**  
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Manager:**  
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Business Owner:**  
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Recommendation**

☐ **Approve for Production** - All critical tests passed, no blocking defects  
☐ **Conditional Approval** - Minor issues remain, workarounds documented  
☐ **Not Ready** - Critical defects exist, additional testing required

**Comments:**

**Appendix J: Known Limitations and Considerations**

**Legacy System Limitations Carried Forward**

1. **Array Size Limitation**: Original system limited to 30,000 TINs per execution
   * **Recommendation**: Java implementation should use dynamic collections (ArrayList) to remove this limitation
   * **Test Verification**: TC\_SIA\_BVA\_007 and TC\_SIA\_BVA\_008
2. **Service Center Code Restriction**: Only 10 specific SC codes supported
   * **Business Rule**: Codes limited to 7, 8, 9, 17, 18, 19, 28, 29, 49, 89
   * **Future Enhancement**: Consider making this configurable
3. **Date Format Requirement**: LOADDT must be MM/DD/YYYY format
   * **Current**: Manual date entry required
   * **Enhancement**: Support ISO 8601 format (YYYY-MM-DD)
4. **Fixed-Width File Format**: Output files use fixed-width format (legacy requirement)
   * **Current**: Maintains compatibility with downstream systems
   * **Future**: Consider CSV or XML option for flexibility

**Java Implementation Considerations**

1. **Character Encoding**: Ensure consistent UTF-8 or ASCII encoding
2. **Thread Safety**: Current design appears single-threaded; evaluate multi-threading for performance
3. **Connection Pooling**: Implement proper connection pool management
4. **Transaction Management**: Clarify commit strategy (batch vs. record-level)
5. **Logging Framework**: Use standard logging (Log4j, SLF4J) for better diagnostics
6. **Configuration Management**: Externalize all configuration (database, file paths, etc.)
7. **Error Recovery**: Implement checkpoint/restart capability for large datasets
8. **Monitoring**: Add JMX beans for runtime monitoring

**Testing Constraints**

1. **Production Data**: Cannot use actual production data for testing due to PII concerns
2. **Performance Testing**: Test environment may not match production hardware specifications
3. **Load Testing**: Concurrent user testing not applicable (batch process)
4. **Time-Based Testing**: Weekend processing timing issue (TC030 posting delay) difficult to simulate

**Appendix K: Glossary of Terms**

| **Term** | **Definition** |
| --- | --- |
| **AEKB** | Advanced Encryption Key Block - encryption method used for password protection |
| **AOTO** | Account Origin Territory - combination of area and territory codes |
| **ASSN** | Assignment table in ALS schema containing taxpayer assignment records |
| **BVA** | Boundary Value Analysis - testing technique focusing on limits of input ranges |
| **BFEUP** | BMF/EPEF/NMF Extract Update Program - transaction type for certain financial systems |
| **BMF** | Business Master File (fs=2) |
| **CLC** | Campus Locator Code - derived from first 2 digits of AOTO |
| **COREDIAL** | Core Dial table containing TIN cross-reference data |
| **CORESID** | Core System Identifier used to determine transaction modification type |
| **DOMAP** | Domain Map table linking Service Centers to geographical areas |
| **EPEF** | Employee Plans Examination File (fs=3) |
| **FS** | Financial System code (1=IMF, 2=BMF, 3=EPEF, 4=IRAF, 6=NMF) |
| **IFEUP** | IMF/IRAF Extract Update Program - transaction type for IMF/IRAF systems |
| **IMF** | Individual Master File (fs=1) |
| **IRAF** | Information Returns Analysis File (fs=4) |
| **LOADDT** | Load Date - date when record was loaded or last processed |
| **NMCTRL** | Name Control - abbreviated taxpayer name identifier |
| **NMF** | Non-Master File (fs=6) - requires ULC Code |
| **NULLDATE** | Default date value '01/01/1900' indicating unprocessed records |
| **QPICKIND** | Queue Pick Indicator (3=eligible, 4=processed) |
| **RONUM** | Routing Number - combination of group and routing numbers |
| **ROMNUM** | ROM Number - typically 1 for primary records |
| **SC** | Service Center - IRS processing center code |
| **SEL\_DATE** | Selection Date - date record was selected for processing |
| **SIA** | Service Identifier Area - system component name |
| **TC015** | Transaction Code 015 - area change without SC change |
| **TC030** | Transaction Code 030 - both area and SC change |
| **TIN** | Taxpayer Identification Number |
| **TRANSMOD** | Transaction Modification flag indicating eligibility (0,1,2,3) |
| **TSIGN** | Transaction Sign - standard transaction without area/SC change |
| **TT** | Transaction Type code |
| **ULC\_CD** | Unit Locator Code - required only for NMF (fs=6) |
| **UPDTFLAG** | Update Flag - controls database modification (Y=update, N=read-only) |

**Appendix L: Reference Documentation**

**Related Documents**

1. **Legacy System Documentation**
   * Original Pro\*C source code: siaextract.pc
   * Database schema documentation
   * IRS processing guidelines
2. **Java Implementation Documentation**
   * Java Design Document
   * API Documentation (JavaDoc)
   * Deployment Guide
   * Operations Manual
3. **Business Requirements**
   * SIA Extract Functional Requirements Specification
   * Business Process Documentation
   * Data Dictionary

**External References**

1. **Oracle Documentation**
   * Oracle Database SQL Reference
   * JDBC Developer's Guide
   * Oracle Performance Tuning Guide
2. **Java Resources**
   * Java SE Documentation
   * JDBC Best Practices
   * JUnit Testing Framework
3. **Testing Standards**
   * IEEE 829 Test Documentation Standard
   * ISTQB Testing Glossary
   * NIST Software Testing Guidelines

**Document Revision History**

| **Version** | **Date** | **Author** | **Changes** |
| --- | --- | --- | --- |
| 1.0 | 09/29/2025 | QA Team | Initial document creation based on legacy system analysis |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

This Test Case Document has been reviewed and approved for use in testing the SIA Extract Java implementation.

**Prepared By:**  
QA Engineering Team  
Date: September 29, 2025

**Reviewed By:**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (QA Manager)  
Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Approved By:**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Project Manager)  
Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**END OF DOCUMENT**

**Quick Reference Card**

**Essential Commands**

# Display version

java -jar sia-extract.jar -version

# Extract single SC with database update

java -jar sia-extract.jar -fix 07 09/29/2025 Y

# Extract all SCs in read-only mode

java -jar sia-extract.jar -fix 0 09/29/2025 N

# Extract specific SC with NULLDATE

java -jar sia-extract.jar -fix 07 01/01/1900 Y

**Valid Service Center Codes**

7, 8, 9, 17, 18, 19, 28, 29, 49, 89

**Financial System Codes**

* 1 = IMF
* 2 = BMF
* 3 = EPEF
* 4 = IRAF
* 6 = NMF (requires ULC)

**Transaction Types**

* trans\_mod = 0: Skip (not eligible)
* trans\_mod = 1: TSIGN
* trans\_mod = 2: TC015 (IFEUP/BFEUP)
* trans\_mod = 3: TC030 (IFEUP/BFEUP)

**Quick Test Verification**

# Check output file created

ls -l SC07/alsent.07.20250929

# Count records in output

wc -l SC07/alsent.07.20250929

# Verify database updates

sqlplus als/password @verify\_updates.sql

# Check error log

tail -f /var/log/sia-extract/error.log

**This document is ready for use in Microsoft Word format. Copy the entire content into a Word document, apply appropriate formatting (headers, tables, page breaks), and save for distribution to your testing team.**