**Excel Report Validator Documentation**

1. Document Properties

| **Category** | **Details** |
| --- | --- |
| Author: | Sukhpreet Saini |
| Last Updated: | 13 August 2025 |
| Version: | 2.0 |
| Software Required: | Python 3.8+, Pandas 1.3+, Microsoft Word |

2. Table of Contents

1. Overview
2. System Requirements
3. Function Specifications
   * 3.1 Cover Page Validation
   * 3.2 Column Structure Testing
   * 3.3 Summary Calculations
4. Installation Guide
5. Usage Examples
6. Error Handling
7. Appendix

3. Overview

This tool provides automated validation for Excel reports through three test suites:

A diagram of a match

AI-generated content may be incorrect.  
*Validation Workflow*

4. System Requirements

4.1 Software

* Python 3.8+
* Pandas 1.3+
* OpenPyXL

4.2 Hardware

* Minimum 4GB RAM
* 500MB Disk Space

5. Function Specifications

5.1 test\_cover\_page(report\_path, expected\_version)

**Purpose**: Validates report metadata and version information

**Parameters:**

| **Parameter** | **Data Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| report\_path | String | Yes | Excel file path |
| expected\_version | String | Yes | Version number (e.g. "1.5") |

**Returns:**

**```**python

{

'title\_spelling': {

'passed': bool,

'message': str

},

'etl\_dates': {

'passed': bool,

'message': str

},

'version': {

'passed': bool,

'message': str

}

}

**```**

**Logic Table:**

1. Extracts text from cover sheet
2. Validates against expected title formats:
   * "Resource Providers Available"
   * "Data accurate as of last successful ETL run"
3. Verifies ETL date sequence:

```python

"ETL - Started: [date]; CM - Completed: [date]"

```

# Ensures start < completion

1. Checks version string format:

```python

r"Version: (\d+\.\d+)" # Matches version pattern

```

| **Test Case** | **Validation Method** |
| --- | --- |
| Title Spelling | Exact string match |
| ETL Dates | Chronological validation |
| Version Number | Regex pattern matching |

5.2 test\_standard\_report\_columns()

**Purpose**: Audits column structure against design specs

**Parameters**:

| **Parameter** | **Default** | **Description** |
| --- | --- | --- |
| design\_header\_row | 7 | CSV row containing column headers |
| report\_sheet\_name | 1 | Excel sheet index (0-based) |
| report\_header\_row | 2 | Excel row containing headers |

A diagram of a match

AI-generated content may be incorrect.**Comparison Logic**:

5.3 test\_summary\_calculations(report\_path)

**Purpose**: Validates summary metrics against calculated values

**Data Sources**:

| **Metric** | **Excel Location** | **Sample Value** |
| --- | --- | --- |
| Brought Forward | Sheet3!B3 | 206 |
| Approved | Sheet3!B4 | 144 |
| End of Period | Sheet3!B6 | 149 |

**Calculation Formula:**

assert (BroughtForward + Approved) - Closed == EndOfPeriod

5.4 run\_all\_tests()

**Test Sequence**:

1. Cover Page Validation
2. Column Structure Audit
3. Summary Calculation Check

**Output Example**:

```text

=== Summary Tests ===

BROUGHT\_FORWARD: PASSED ✅

Expected: 206 (from B3), Actual: 206

APPROVED: PASSED ✅

Expected: 144 (from B4), Actual: 144

END\_OF\_PERIOD: PASSED ✅

Calculation: 206 + 144 - 201 = 149 ✔️

```

6. Installation Guide

1. Clone repository:

```bash

git clone <https://github.com/SUKH2022/Automation_Testing.git>

```

1. Install dependencies:

```bash

pip install -r requirements.txt

```

7. Usage Example

```python

# Import module

from validator import run\_all\_tests

# Run test suite

run\_all\_tests(

report\_path="report.xlsx",

design\_spec\_path="design.csv",

expected\_version="1.5"

)

```

8. Error Handling Matrix

| **Error Type** | **Handling Method** |
| --- | --- |
| File Not Found | Returns detailed path error |
| Invalid Date Format | Flags unparseable dates |
| Type Conversion | Validates numeric values |
| Missing Columns | Identifies absent fields |
| Sheet Access | Handles sheet index errors |

A pie chart with numbers and text

AI-generated content may be incorrect.