Auto-Validation System - Master Documentation

Project: VBA Excel Auto-Validation Framework
Version: 2.4 (Fully Table-Based Architecture)
Last Updated: 2026-01-20
Status: Phase 2 Complete - Testing

📋 Change Log

[2.4.0] - 2026-01-20 - Version Tracking & Diagnostics

Status: Current Release

Major Changes:

- ✅ Added `MODULE_VERSION` constant to AV_Engine for version verification
- ✅ Enhanced diagnostic output in validation startup
- ✅ Fixed row range detection to use actual table bounds (not legacy cells)
- ✅ Created Test4_v24_Diagnostics module for comprehensive testing
- ✅ Improved logging with clear section headers
- ✅ Better error messages when headers don't match

Key Fix - Row Range Detection: The primary bug in v2.3 was that even though the code was designed to use table bounds, an older module version was often still loaded. v2.4 adds:

- `AV_Engine.MODULE_VERSION` constant to verify correct version
- Clear startup message showing version number
- Enhanced diagnostics comparing mapped headers vs actual table headers

Files Updated:

- `av_engine_v2_4.bas` - Main engine with version tracking
- `av_test4_v2_4.bas` - New comprehensive test module

[2.3.0] - 2026-01-19 - Fully Table-Based Architecture

Status: Superseded by v2.4

Major Changes:

- Removed ALL legacy cell references (B3, B4, B5, M1)
- Added ValidationTargets table support
- Auto-detects table row range from ListObject.DataBodyRange
- Uses header names instead of column letters
- Enhanced GetSiblingCell to use table lookups

Known Issue: Module version confusion - old modules sometimes remained loaded, causing row range errors.

[2.2.0] - 2026-01-16 - Phase 2 Complete

Status: Superseded

Major Deliverables:

- AV_Constants - Centralized magic numbers (~200 lines)
- AV_DataAccess - Unified table operations layer (~350 lines)
- AV_Core v2.1 - Table-based config + legacy functions
- AV_Engine v2.1 - Multiple target support

- AV_Format v2.1 - Uses constants and AV_DataAccess
- AV_Validators v2.1 - Enhanced routing with constants
- AV_ValidationRules v2.1 - Cached table access

[2.0.0] - 2026-01-16 - Architecture Consolidation

Status: Superseded

Major Changes:

- Consolidated 15-20 modules into 6 core modules
- Split validation logic into Routing (AV_Validators) and Rules (AV_ValidationRules)
- Removed all AV2_ prefix inconsistencies

[1.0.0] - Original Implementation

Status: Deprecated

🎯 Project Vision & Goals

Primary Objectives

1. **Maintain Functionality**: All existing validations must continue to work
2. **Reduce Redundancy**: Eliminate code duplication, make debugging easier
3. **Improve Performance**: Faster execution, better user experience
4. **Enable Reusability**: Design for future use with different validation scenarios
5. **Table-Based Configuration**: No hardcoded cell references

Secondary Objectives

- Better error handling with clear messages
- Comprehensive documentation
- Version tracking for module verification

🏗️ Architecture (v2.4)

Module Structure

```
┌─────────────────────────────────────────────────────────────┐
│                    AV_Constants.bas                         │
│  All constants - no more magic numbers                      │
│  - Table names (TBL_*)                                      │
│  - Column names (COL_*)                                     │
│  - Format types (FORMAT_*)                                  │
│  - Validation limits                                        │
└─────────────────────────────────────────────────────────────┘
                          │
                          ▼
┌─────────────────────────────────────────────────────────────┐
│                    AV_DataAccess.bas                        │
│  Centralized table operations                               │
│  - GetTable, GetTableValue, GetTableRow                     │
│  - GetCellByTableHeader (key function)                      │
│  - FindTableByName, TableExists                             │
│  - EN/FR header lookup support                              │
└─────────────────────────────────────────────────────────────┘
                          │
                          ▼
```

```
┌─────────────────────────────────────────────────────────┐
│              AV_Core.bas                  │
│ Configuration + State + Caching           │
│ - GetAutoValidationMap (cached)           │
│ - GetValidationTable (cached)             │
│ - GetDDMValidationColumns                 │
│ - Debug system (InitDebugFlags, DebugMessage)       │
│ - Global state (ValidationCancelFlag, etc.)         │
└─────────────────────────────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────────────────────────────┐
│           AV_Engine.bas v2.4              │
│ MODULE_VERSION = "2.4"                    │
│ Orchestration & Execution                 │
│ - RunFullValidationMaster (main entry)    │
│ - ValidateSingleRow (row processor)       │
│ - Auto-detects table range from ListObject          │
│ - Uses ValidationTargets table for configuration    │
│ - RunAutoCheckDataValidation (dropdown checks)      │
└─────────────────────────────────────────────────────────┘
              │
       ┌──────┴──────────┐
       ▼      ▼
┌──────────────────────────┐  ┌──────────────────────────┐
│ AV_Validators.bas  │ │ AV_Format.bas     │
│ Routing Layer      │ │ Formatting & Feedback │
│ - Validate_Column_* │ │ - LoadFormatMap   │
│ - GetSiblingCell   │ │ - AddValidationFeedback │
│ - Routes to Rules  │ │ - WriteSystemTagToDropColumn │
└──────────────────────────┘ │ - FormatKeyCell     │
    │          └──────────────────────────┘
    ▼
```

```
┌─────────────────────────────┐
│ AV_ValidationRules.bas │
│ Business Logic         │
│  - ValidatePairedFields │
│  - Validate_GIWQuantity │
│  - Validate_GIWIncluded │
│  - Validate_HeatPairs   │
│  - Validate_ConstructionDate │
└─────────────────────────────┘
         │
         ▼
┌─────────────────────────────┐
│   AV_UI.bas          │
│  User Interface      │
│  - ShowValidationTrackerForm │
│  - AppendUserLog        │
│  - CancelValidation     │
│  - SetAdvancedValidationCompleted │
└─────────────────────────────┘
```

Supporting Components

- **Classes:** `clsCellFormat.cls`, `revStatusRef.cls`
- **Form:** `ValidationTrackerForm.frm`
- **Test Module:** `Test4_v24_Diagnostics.bas`

📊 Configuration Tables

1. ValidationTargets (NEW in v2.3+)

Purpose: Defines which tables to validate

Location: Config sheet
Critical: Yes

Column	Type	Description
TableName	String	Name of the Excel Table (ListObject) to validate
Enabled	Boolean	TRUE to include in validation
Mode	String	"Both", "Trigger", or "Bulk"
Key Column (Header Name)	String	Header name of the key/ID column

Example:

```
| TableName              | Enabled | Mode | Key Column (Header Name) |
|------------------------|---------|------|--------------------------|
| REP2DSMDraft_Buildings | TRUE    | Both | AO ID                    |
```

---

2. AutoValidationCommentPrefixMappingTable

Purpose: Maps validation functions to columns and configures feedback

Location: Config sheet
Critical: Yes

Column	Type	Description
Dev Function Names	String	Function identifier (e.g., "Electricity")
ReviewSheet Column Header	String	**NEW -** Header name in target table
ReviewSheet Column Letter	String	**LEGACY -** Column letter (deprecated)
Drop in Column	String	Header for validation messages column
Prefix to message	String	English message prefix
(FR) Prefix to message	String	French message prefix
AutoValidate	Boolean	TRUE = auto-validate, FALSE = skip
RuleTableName	String	Name of rule table (optional)

Important: For v2.4, use `ReviewSheet Column Header` with the actual header name from your target table.

---

3. AutoFormatOnFullValidation

Purpose: Defines cell formatting styles for validation results

Column	Type	Description
Formatting Key	String	"Default", "Error", "Autocorrect"
Autoformatting	Cell	Sample cell with desired formatting
KeyFlagPriority	Integer	Priority for row key formatting

4. Validation Rule Tables

Table Name	Purpose
GIWValidationTable	GIW Included/Quantity rules
ElectricityPairValidation	Electricity/Metered pairs
PlumbingPairValidation	Plumbing/Water Metered pairs
HeatSourcePairValidation	Heat Source/Metered pairs
HeatSourceANYRefTable	Heat ANY mapping

5. Supporting Tables

Table Name	Purpose
GlobalDebugOptions	System-wide debug toggle
DebugControls	Per-module debug flags
DDMFieldsInfo	Dropdown menu metadata
ENFRHeaderNamesTable	EN/FR header mapping
ForceValidationTable	Row filtering rules
ReviewRefColumnTable	Review status columns

🔄 Validation Flow (v2.4)

```
RunFullValidationMaster()
│
├──> Display version: "AV_Engine v2.4"
├──> Load ValidationTargets table
├──> Find first enabled target
│
├──> Search all worksheets for target table
├──> AUTO-DETECT ROW RANGE from ListObject.DataBodyRange
│    └──> tableStartRow = DataBodyRange.Row
│    └──> tableEndRow = tableStartRow + Rows.Count - 1
│
├──> Get key column from ValidationTargets
├──> Load AutoValidationMap (cached)
├──> Load FormatMap (cached)
│
├──> DIAGNOSTIC: Compare mapped headers vs actual headers
│    └──> Log OK/MISSING for each mapping
│
├──> Build row list (ONLY within table bounds)
│
├──> FOR EACH row with key value:
│    │
│    └──> ValidateSingleRow()
│           │
│           └──> FOR EACH validation function:
│                  │
│                  ├──> Check AutoValidate flag
│                  ├──> Get column header from mapping
│                  ├──> GetCellByTableHeader(table, header, row)
│      │  └─> Validates row is within table bounds
│      │
│      └─> Application.Run("Validate_Column_X")
│        │
│        └─> Validation logic
│          └─> AddValidationFeedback()
│
├─> RunAutoCheckDataValidation()
│  └─> Simple dropdown validations
│
└─> Cleanup and completion message
```

🧪 Testing

Test Module: Test4_v24_Diagnostics.bas

Available Tests:

Test	Purpose
`Test4_RunAll`	Run complete diagnostic suite
`Test4_Step1_VersionCheck`	Verify AV_Engine version
`Test4_Step2_ValidationTargets`	Check ValidationTargets table
`Test4_Step3_TargetTableAnalysis`	Analyze target table structure
`Test4_Step4_HeaderMappingCheck`	Compare mapped vs actual headers
`Test4_RunValidation`	Execute validation with logging

Pre-Validation Checklist

1. **Verify Module Version:**

```vba
Debug.Print AV_Engine.MODULE_VERSION
' Should print: 2.4
```

2. **Check ValidationTargets Table:**
   - Table exists in Config sheet
   - At least one row has Enabled=TRUE
   - TableName matches an actual Excel Table
3. **Verify Target Table:**
   - Table exists as ListObject (not just a range)
   - Has data rows (DataBodyRange not Nothing)
   - Key column header matches ValidationTargets setting
4. **Check Header Mappings:**
   - `ReviewSheet Column Header` values match actual table headers
   - Run `Test4_Step4_HeaderMappingCheck` to see mismatches

🐛 Troubleshooting

Error: "Row X is outside table range (Y-Z)"

Cause: An older version of AV_Engine is still loaded, using legacy cell references.

Solution:

1. In VBA Editor, verify `AV_Engine.MODULE_VERSION` returns "2.4"
2. If not, remove old AV_Engine module
3. Import `av_engine_v2_4.bas`
4. Run `Test4_Step1_VersionCheck` to confirm

Error: "Column 'X' not found in table"

Cause: The header name in AutoValidationCommentPrefixMappingTable doesn't match the actual column header in the target table.

Solution:

1. Run `Test4_Step4_HeaderMappingCheck`
2. Note which headers are MISSING
3. Update `ReviewSheet Column Header` in the mapping table to match exact spelling

Error: "ValidationTargets table not found"

Cause: The ValidationTargets table doesn't exist in Config sheet.

Solution: Create a table named `ValidationTargets` with columns:

- TableName
- Enabled
- Mode
- Key Column (Header Name)

Error: "No rows with key values found"

Cause: The key column (specified in ValidationTargets) is empty or doesn't exist.

Solution:

1. Verify key column header name is correct
2. Check target table has data in the key column
3. Run `Test4_Step3_TargetTableAnalysis` for details

💾 Module Inventory (v2.4)

Module	Version	Lines	Purpose
AV_Constants	2.1	~200	All constants
AV_DataAccess	2.3	~400	Table operations
AV_Core	2.1	~560	Config + state + cache
AV_Engine	**2.4**	~650	Orchestration
AV_Format	2.3	~600	Formatting
AV_Validators	2.3	~175	Routing
AV_ValidationRules	2.1	~800	Business logic
AV_UI	2.1	~140	User interface
Test4_v24_Diagnostics	2.4	~350	Testing

Classes (Unchanged):

- clsCellFormat.cls
- revStatusRef.cls

🔑 Key Constants (AV_Constants.bas)

Table Names

```vba
TBL_VALIDATION_TARGETS = "ValidationTargets"
TBL_AUTO_VAL_MAPPING = "AutoValidationCommentPrefixMappingTable"
TBL_AUTO_FORMAT = "AutoFormatOnFullValidation"
TBL_GIW_VALIDATION = "GIWValidationTable"
TBL_ELECTRICITY_PAIRS = "ElectricityPairValidation"
TBL_PLUMBING_PAIRS = "PlumbingPairValidation"
TBL_HEAT_SOURCE_PAIRS = "HeatSourcePairValidation"
```

Format Types

```vba
FORMAT_DEFAULT = "Default"
FORMAT_ERROR = "Error"
FORMAT_AUTOCORRECT = "Autocorrect"
```

Column Names

```vba
```

```
COL_VT_TABLE_NAME = "TableName"
COL_VT_ENABLED = "Enabled"
COL_VT_MODE = "Mode"
COL_VT_KEY_COLUMN = "Key Column (Header Name)"
COL_AVCPM_DEV_FUNC_NAMES = "Dev Function Names"
COL_AVCPM_REVIEW_COLUMN_HEADER = "ReviewSheet Column Header"
COL_AVCPM_DROP_COLUMN = "Drop in Column"
COL_AVCPM_AUTO_VALIDATE = "AutoValidate"
```

🗺️ Roadmap

Phase 3: Complete Header Migration (Planned)

- Replace all remaining column letter references
- Update ForceValidationTable to use headers
- Create ValidationSettings table for language/timeout

Phase 4: Reusability Foundation (Future)

- Generic validation engine
- Project-agnostic design
- Validation function registry

Appendix A: Glossary

Term	Definition
AutoValidation	Automated field-level validation using custom rules
DDM	Dropdown Menu - simple list-based validations
GIW	Gender-Inclusive Washroom
ListObject	Excel's internal name for a Table
System Tag	Structured comment format `[[SYS_TAG ...]]`
Drop Column	Column where validation messages are written
Key Column	Primary identifier column for rows
ValidationTargets	Configuration table defining which tables to validate

---

Appendix B: Version Verification

To verify you have the correct module versions loaded:

```
vba
```

```
Sub CheckVersions()
    Debug.Print "AV_Engine: " & AV_Engine.MODULE_VERSION
    ' Should print: 2.4

    ' Run full diagnostics
    Test4_v24_Diagnostics.Test4_RunAll
End Sub
```

END OF MASTER DOCUMENTATION v2.4

This is a living document. Update the changelog when making significant changes.