FactSet Concordance Rules for Security Identifier Mapping

Category: Technical

Model: BDWP-003

Table of Contents

1. Introduction

This technical document defines the **FactSet concordance rules** used for mapping security identifiers—specifically CUSIP, ISIN, and SEDOL—within FactSet's data ecosystem. These rules ensure consistent and accurate linkages across vendor data sources, facilitate efficient data integration, and support automated validation and reconciliation processes. The document also elaborates on handling variant formats, fallback strategies, exception cases, and updates following schema alterations or emerging data patterns.

These rules underpin core modules such as ADR mapping, ETF constituent reconciliation, and vendor taxonomy updates, providing critical guidance for validation, troubleshooting, and steward interventions.

2. Scope and Purpose

This document covers:

- Standardized mapping logic for CUSIP, ISIN, and SEDOL identifiers within FactSet systems.
- Procedures for handling variant formats and vendor-specific suffixes.
- Fallback and override mechanisms in cases of ambiguous or conflicting identifiers.
- Validation checklists for ensuring conformity, accuracy, and completeness of mappings.
- Procedures for updating rules following vendor schema changes or new data patterns.

The primary purpose is to support data integrity, facilitate automation, and provide clear guidelines for stewards and technical teams involved in security identifier management.

3. FactSet Concordance Rules

3.1 Overview

FactSet's concordance rules establish the logic for linking various identifier

formats to unified security entities. They leverage hierarchical and logical patterns, prioritization matrices, and vendor-specific considerations to optimize accuracy.

3.2 Key Logic Principles

- **Consistency:** Use standard formats (e.g., 9-character CUSIP, 12-character ISIN, 6-character SEDOL).
- **Normalization:** Strip extraneous characters, whitespace, and vendor suffixes before matching.
- **Prioritization:** Preference order for data sources—FactSet internal mapping overrides vendor data unless explicitly overridden.
- **Fallback Handling:** When primary identifiers are missing or incompatible, use linked identifiers or last-resort hashes.

3.3 Example Mapping Logic

> **Scenario:** Mapping a security with both CUSIP and ISIN identifiers provided.
>
> Rule: *Verify consistency between CUSIP and ISIN using cross-validation. Prioritize CUSIP if available; cross-check with ISIN for conflicts. If mismatch detected, escalate for stewardship review or apply override if approved.*

4. Handling Variant Identifier Formats

4.1 Vendor Suffixes and Variants

Identifiers often include suffixes denoting vendor or market origin, e.g., `.LN` (London), `.GR` (Germany), or `.HK` (Hong Kong). These suffixes are non-essential for core mapping but should be normalized.

4.2 Normalization Procedures

1. Strip suffixes: Example: `123456789.LN` → `123456789`
2. Convert to uppercase: Ensures uniformity regardless of original casing.
3. Remove extraneous characters such as hyphens, spaces, or special symbols.

4.3 Example

> Identifier before normalization: `US0378331005.GH`
>
> After normalization: `US0378331005`

5. Fallback and Exception Procedures

5.1 Primary and Secondary Identifiers

When primary identifiers fail validation or conflict arises, fallback procedures are employed:

- Use linked identifiers (e.g., LEI, proprietary keys).
- Check alternate data sources for corroboration.
- Apply last-resort hashes for unmatched entities.

5.2 Exception Handling and Overrides

Overrides require stewardship approval based on an established matrix (see section 10). Exceptions are documented via lineage forms and must be reviewed periodically.

5.3 Example Troubleshooting Flow

> **Issue:** Conflicting CUSIP and ISIN for a security.
>
> 1. Verify identifier formats and normalization.
> 2. Check vendor suffixes and adjust accordingly.
> 3. Review hierarchy of data sources for priority.
> 4. Consult stewardship override procedures if conflict persists.

6. Validation Checklists

6.1 Checklist Objectives

- Ensure formatting conforms to standardized patterns.
- Confirm absence of extraneous characters or suffixes.
- Verify cross-source consistency between identifiers.
- Check for duplication or conflicts within the system.

6.2 Sample Validation Checklist

Step	Activity	Pass/Fail Criteria
1	Validate format pattern	Identifier matches expected regex (*e.g., CUSIP: 9 alphanumeric characters*)
2	Normalize identifier	Suffixes removed; uppercase; no spaces or special characters
3	Cross-validate with linked identifiers	No mismatch detected; encodes same security
4	Check for duplicates	Identifier is unique or correctly associated

Regularly update validation checklists to incorporate new identifier formats or data anomalies.

7. Updating Rules and Procedures

7.1 Triggers for Updates

- Vendor schema changes (e.g., Bloomberg schema 2025-08-19).
- Emergence of new identifier variants or formats.
- Detection of recurring validation conflicts or errors.
- Regulatory or compliance requirements update.

7.2 Update Process

1. Monitor vendor notifications and schema releases.
2. Assess impact on existing mappings via impact analysis.
3. Revise normalization and validation logic accordingly.
4. Update validation checklists and stewardship directives.
5. Document changes with version control and lineage logs.
6. Communicate updates to relevant teams and stakeholders.

7.3 Version Control and Documentation

All rule modifications must be logged in the version control system, and change approvals must be tracked via stewardship sign-off documentation.

8. Error Codes and Troubleshooting

8.1 List of Common Error Codes

Error Code	Description	Symptoms	Root Causes	Resolution Steps

MDM-ID-3102	Identifier conflict	Multiple identifiers mapped to same security.	Duplicate security entries or conflicting vendor data.	Validate identifier formats; reconcile duplicates; escalate for stewardship review if unresolved.
MDM-LIN-2207	Lineage gap	Missing historical linkage data.	Data integration issue, missing source record.	Review source feed; verify data completeness; add lineage entries to close gap.
MDM-COM-1905	Commentary required	Flagged when manual intervention needed.	Data uncertainty or known conflicts.	Review comment; escalate for steward input; document resolution.

8.2 Troubleshooting Workflow

> **Step 1:** Identify error code from system logs or validation reports.
>
> **Step 2:** Refer to theerror code details table for root causes and resolution steps.
>
> **Step 3:** Execute resolution steps, such as re-normalizing identifiers or validating cross-references.
>
> **Step 4:** Confirm resolution through re-validation or manual review.

9. Module and Data Coverage

9.1 Affected Modules

- **ADR Mapping:** Ensures accurate ADR security linkages.
- **ETF Constituent Reconciliation:** Validates ETF component identifiers.
- **Vendor Taxonomy Updates:** Incorporates schema changes like Bloomberg 2025-08-19 and Moody's codes.

9.2 Data Patterns and Vendor Schemas

Schema / Pattern	Description	Impacted Identifiers
Bloomberg 2025-08-19	Vendor-specific schema with suffix patterns	CUSIP, ISIN, SEDOL with suffixes (.LN, .GR, .HK)
Moody's Sector Codes	Classification codes such as MDY-SEC-24xx	Sector and industry identifiers linked to security IDs

10. Steward Override Procedures and Approval Matrix

Overrides are permissible only under documented circumstances. All overrides require adherence to an approval matrix based on severity, impact, and data sensitivity:

Override Type	Approvers	Documentation Required	Approval Level

Minor Conflict Resolution	Data Steward, Technical Lead	Override request form, rationale, supporting data	Level 1
Major Override (Schema change impact)	Data Governance Committee	Comprehensive impact analysis and approval	Level 2

10.1 Override Logging and Tracking

All overrides must be logged in the stewardship oversight system, referencing the lineage documentation template (see section 11). Periodic audits are conducted to review exception cases and validate ongoing compliance.

11. Lineage Documentation Templates

Lineage templates are used to document data flow, transformations, and overrides. They capture:

- Source schema and data version
- Normalization procedures applied
- Override details and steward comments
- Version history and change justifications

11.1 Example Lineage Template Fields

Field	Description