XBRL GL Next Taxonomy Framework— Part 5: Tavi Palette Taxonomy

Document status: Draft for Collaborative Review
Date: **2026-11-06**
Author: Nobuyuki SAMBUICHI
Role: Advisor to XBRL Japan
Project: XBRL GL Next
Distribution: Public collaborative review draft.
Repository version: public collaborative review draft.
Editors: SAMBUICHI, Nobuyuki
Contributors: invited project participants (individual names not yet confirmed for publication).

Status

This document is a draft prepared by Nobuyuki SAMBUICHI, Advisor to XBRL Japan, as part of the XBRL GL Next project. It is made available through a public GitHub repository for collaborative review. It has not been approved or endorsed as an official specification by XBRL Japan, XBRL Europe, XBRL International, or any other organisation. Copyright and licence terms remain under review.

The requirements in this Part describe a proposed XBRL GL Next binding to Project Tavi; they do not amend the external Tavi specification or establish processor conformance. This candidate revision is not an approved Canonical or Official GIT release.

Abstract

This Part describes the proposed XBRL GL Next binding to Project Tavi, based on a purpose-specific HMD View derived from an approved HMD for taxonomy. It defines the controlled 14-column view contract and four independently derived definition tables for Concepts and Datatypes, Dimensions, Presentation and Labels, together with their mapping to the selected Tavi Public Working Draft JSON model.

The proposed realisation preserves source semantic paths, selected facts and occurrence relationships while allowing singular Class folding and repeating-Class dimensions under the stated view rules. It separates the taxonomy, embedded Table Template, Instance Module and Structured Tidy Data CSV and specifies the evidence required to assess their correspondence.

The published integer-key Journal Entry example is informative evidence of a candidate implementation, not proof of complete derivation from an approved HMD View or formal Tavi conformance. External Tavi specification authority, unresolved module usage and incomplete native report validation remain explicitly separate from the XBRL GL Next proposal.

Table of Contents

Status 1
Abstract 1
1 Language independence 4
2 Introduction 4
2.1 Scope 4
2.2 Normative language 4
2.3 Relationship to the other Parts 4
2.4 Implementation independence 5
3 Terms and definitions 5
4 Definition order 6
4.1 Overview 6
4.2 Row validation 7
4.3 Non-use subtrees and restrictions 7
5 Source and HMD View contracts 7
6 QName construction for Tavi 7
7 Four independent definition tables 8
8 Recursive processing for the Dimension Definition Table 8
9 Independent table contracts 9
10 Tavi Taxonomy JSON Mapping 10
10.1 Mapping scope, source fields and naming 10
10.2 Concepts and Datatype Definition Table (3a) 10
10.3 Dimension Definition Table (3b) 11
10.4 Presentation Definition Table (3c) 13
10.5 Label Definition Table (3d) 13
10.6 JSON properties not supplied directly by the four tables 14
10.7 Informative COR Journal Entry trace 15
11 Tavi binding model 15
11.1 Attribute facts 15
11.2 Structural rows 15
11.3 Class resources 15
11.4 Native networks 16
12 Dimensional network generation 16
12.1 Cube fact scope 16
12.2 Occurrence dimensions 16
12.3 Attribute membership 16
12.4 Descendant Classes and native relationships 16
12.5 Reading the dimensional hierarchy 16
12.6 Multiplicity 17
12.7 Integer-key PoC and restricted-root alternative 17
13 Tavi taxonomy artefacts 17
13.1 Native taxonomy document 17
13.2 Labels and documentation 17
13.3 Hierarchy network 17
13.4 Network set 18
13.5 View entry point and DTS separation 18
13.6 Domains and cubes 18
13.7 Discovery and dependency resolution 18

14 Instance Module and Structured Tidy Data CSV **18**
14.1 Separation from taxonomy generation **18**
14.2 documentInfo **19**
14.3 Occurrence key columns **19**
14.4 Property columns **19**
14.5 One Structured CSV per report **19**
14.6 Reading a row through the cube **19**
14.7 Report validation **19**
14.8 Shared semantics and Value Domains **20**
14.9 Implementation profile and evidence **20**
15 Cross binding equivalence **20**
16 Accounting Entries baseline and additional HMDs **20**
17 Formal package assembly **20**
17.1 Package layout **20**
17.2 Version and namespaces **21**
17.3 Deterministic view identifier **21**
17.4 Publication boundary **21**
18 Validation and conformance **21**
18.1 Taxonomy conformance **21**
18.2 Minimum report tests **21**
19 Output name summary **22**
20 Instance examples **22**
20.1 Example files **22**
20.2 Accounting Entries example **22**
20.3 Additional views **23**
20.4 Concepts dimensions and presentation **23**
20.5 Unit handling and validation status **23**
Appendix A (informative) Tavi taxonomy **24**
A.1 Public integer-key Journal Entry taxonomy: tavi-tax.json **24**
A.2 Public integer-key Journal Entry instance module: tavi-ins.json **25**
A.3 Public integer-key Structured Tidy Data: tavi-ins.csv **25**
Appendix B References **27**
Appendix C Intellectual property status **28**
Appendix D Document history **29**
Appendix E (informative) Correspondence with XBRL GL Next Requirements **30**

1 Language independence

This draft uses UK English. Identifiers, qualified names, namespace URIs, JSON property names and filenames are case-sensitive and are not translated. English and local-language labels and documentation retain the meanings supplied by the accepted model.

2 Introduction

2.1 Scope

This Part governs the Tavi realisation of an HMD for taxonomy view, its reusable taxonomy objects, an explicitly mapped report and its Structured Tidy Data CSV. It does not authorise an alternative semantic model, a new production generator, or a change to the existing Tuple/OIM realisation. The current Tavi PWD 2026-09-01 form uses documentInfo and xbrlModel, not the 2025 OIM Taxonomy documentInfo/taxonomy object form.

2.2 Normative language

The key words "**MUST**", "**MUST NOT**", "**SHOULD**", "**SHOULD NOT**", and "MAY" in this document are to be interpreted as described in BCP 14 (RFC 2119 and RFC 8174) when, and only when, they appear in all capitals.

"**MUST**" and "**MUST NOT**" indicate mandatory requirements and prohibitions, respectively. "**SHOULD**" and "**SHOULD NOT**" indicate recommendations for which deviations may be justified in particular circumstances, provided that their implications are carefully considered. "**MAY**" indicates a permitted option.

The normative or informative status of provisions is determined by the document's stated conventions and the context in which they appear.

2.3 Relationship to the other Parts

The XBRL GL Next Requirements Specification provides the background, design objectives, and requirements for the Framework.

The XBRL GL Next Taxonomy Framework consists of the following Parts:

- XBRL GL Next Taxonomy Framework — Part 1: Semantic Modelling and Taxonomy Foundations;
- XBRL GL Next Taxonomy Framework — Part 2: Tuple Palette Taxonomy;
- XBRL GL Next Taxonomy Framework — Part 3: OIM Palette Taxonomy;
- XBRL GL Next Taxonomy Framework — Part 4: Aligned Pool and Extension Governance;
- XBRL GL Next Taxonomy Framework — Part 5: Tavi Palette Taxonomy.

Part 5 specifies how an HMD View is derived from an approved HMD for taxonomy by selecting the required semantic occurrences and applying permitted multiplicity restrictions. The resulting HMD View provides the common input to four independent definition tables: Concepts and Datatypes, Dimension, Presentation and Label. These tables provide the definitions required for the corresponding components of a Tavi JSON taxonomy. Table Templates subsequently bind Structured Tidy Data to the taxonomy while preserving semantic identity and traceability.

Parts 2, 3 and 5 provide alternative taxonomy realisations of the common semantic framework established by Part 1, under the extension governance defined in Part 4. Part 5 differs in that its Tavi realisation is based on a purpose-specific HMD View rather than necessarily the complete source HMD.

COR, BTX and AAA constitute the reusable semantic paint boxes of XBRL GL Next. COR provides core accounting and ledger semantics, anchored by gl-cor:accountingEntries. BTX provides business-

transaction semantics through its independent root while reusing applicable COR definitions. AAA extends COR with audit and assurance semantics and may reference evidence represented through BTX.

Each purpose-specific HMD serves as a palette, selecting and arranging the required semantics from these paint boxes for a particular business purpose. The resulting taxonomy provides the corresponding technical realisation for use in an individual implementation or dataset.

2.4 Implementation independence

The requirements govern observable results and traceability, not a particular implementation.

A Tavi generator and an HMD-view-to-table-template generator require separately documented authority and versioned interfaces.

The Part 3 reference generator does not automatically become the Tavi generator. Candidate tool placement in canonical/tavi-poc/tools does not by itself approve a model, taxonomy or report.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Part 1, Part 2, Part 3, Part 4 and the following apply. Unless otherwise specified, these terms constitute the common terminology used in this Part and the other Parts of the XBRL GL Next Taxonomy Framework. Compatibility mappings may identify legacy or implementation-specific terms and labels that correspond to these terms.

HMD for taxonomy
Existing accepted root-specific 18-column semantic model source. Its final xpath column is not part of the HMD view input.

HMD View
The selected, reviewed and multiplicity-restricted 14-column semantic input for Tavi taxonomy generation, derived from the accepted 18-column root HMD. Its source sequence, relative row order, level, type and semantic_path are preserved.

Non-use
0..0 selection directive that may exclude even a source-required occurrence; excluded Class subtrees do not contribute Tavi objects

Represented Class
Root navigation anchor or repeating Class represented in the Tavi hierarchy network.

Folded Class
Non-root Class with effective maximum 1: its structure is transparent and its retained Attributes map to the nearest represented ancestor.

Occurrence dimension
Dimension identifying a repeating Class occurrence and inherited by descendant fact space.

Definition table
Logical row-based specification of a Tavi object array or relationship collection, not a requirement to create an additional physical CSV.

Tavi Taxonomy
Model-defining JSON document containing concepts, dimensions, networks, cubes, labels and tableTemplates under xbrlModel.

JSON Map
Per-report Tavi module/metadata JSON that identifies taxonomy, table template, data source, Fact Map and Fact Source; distinct from a JSON Schema.

Instance Module
A Tavi module document that provides the metadata required to associate a specific Structured Tidy

Data CSV with an applicable Tavi taxonomy and its Table Template. It identifies the taxonomy, Table Template and data source through `importMapping`, `importedTaxonomies`, `factMaps`, `factSources` and `sourceMappings`.

The Instance Module serves a purpose analogous to the JSON metadata document in xBRL-CSV, without redefining the imported taxonomy.

Note 1 — A metadata mechanism corresponding to the JSON metadata used in xBRL-CSV is required to associate a Structured Tidy Data CSV with its taxonomy definitions. The Journal Entry PoC uses a Tavi Instance Module for this purpose. Although its JSON structure follows the syntax defined in the Tavi PWD, whether this particular use of a Tavi module is permitted by the specification remains unresolved. Syntactic validity alone does not establish conformance of this usage.

CSV Instance
Physical Structured Tidy Data CSV holding occurrence keys and reported values for the corresponding JSON Map.

Schema validation
A basic structural check that confirms whether a JSON document conforms to the format and structural constraints defined in the externally assigned tavi-schema.json.

4 Definition order

4.1 Overview

The Journal Entry HMD View MUST first be defined by selecting rows from the accepted COR HMD for taxonomy and, where justified by the approved use case, restricting multiplicity. The four definition tables MUST then be derived independently from that reviewed 14-column HMD View. The original 18-column HMD MUST remain unchanged.

Stage	Direct input	Output / responsibility
1. HMD View selection	HMD for taxonomy	Select Journal Entry PoC rows and apply approved multiplicity restrictions to form the use-case View.
2. Approved HMD View	Selected HMD occurrences	Preserve original source sequence, row order, level, type and semantic_path. Fix the effective multiplicities in the 14-column HMD View.
3a. Concepts/Datatype	HMD View only	Define concept QNames, types, datatype bindings and Value Domain references.
3b. Dimension	HMD View only	Remove C/R rows with View multiplicity 1/1..1, recursively reduce descendant levels, and define dimensional scopes.
3c. Presentation	HMD View only	Define presentation anchors, reportable Attribute concepts, parents and sibling order.
3d. Label	HMD View only	Define ordinary and documentation labels in English and Japanese.
4. Tavi JSON	Four independently derived definitions	Serialise assigned Tavi objects using approved fixed binding rules; resolve object references during package validation.

Independence means that none of the four table-generation procedures reads another derived table. It does not prohibit resolved QName references among the final Tavi JSON objects.

4.2 Row validation

Rows MUST match unique source semantic paths. Their module, type, identifier, datatype, local name, definitions, labels and semantic identity MUST match the source. Input order MUST preserve source order. C and R rows have no Attribute datatype; A rows require an approved explicit datatype mapping. Unknown types, unresolved qualified names, conflicting declarations and missing source ancestors are errors. Multiplicity is the only semantic field permitted to differ in the view under this Part; new semantics require upstream modelling and review.

4.3 Non-use subtrees and restrictions

Treat a multiplicity as the interval [minimum, maximum], with * denoting an unbounded maximum. The view interval MUST be contained in the source interval. The view notation 0 means non-use and is interpreted as 0..0. Non-use is a permitted selection directive, including for a source-required occurrence, but makes the view a projection rather than proof of full-model completeness. Non-use excludes the occurrence and its subtree. Restricting 0..* to 0..1 or 1..1, and 1..* to 1..1, is permitted; widening is prohibited. Attributes retained under a selected Class MUST retain their source meaning and applicable requiredness. Omitted required siblings MUST be disclosed in the view review.

5 Source and HMD View contracts

The source COR HMD for taxonomy is the accepted 18-column formal HMD. It is immutable for this derivation. The direct input to all four definitions is the HMD View using the following 14 columns, in the specified order:

```
sequence,module,local_name,level,type,identifier,datatype,multiplicity,value_domain,name
,label_local,definition,definition_local,semantic_path
```

The HMD View preserves the values of sequence, module, local_name, level, type, identifier, datatype, value_domain, name, label_local, definition, definition_local and semantic_path for each selected row. The source row order is preserved, and sequence values are not renumbered. multiplicity is the one controlled exception: a chosen View occurrence may be restricted to a subset of the source occurrence interval, subject to the applicable approval and review rules. The HMD View is not compacted by deleting singular Class nodes; hierarchy reduction belongs exclusively to the derived Dimension Definition Table.

The selected HMD View and its multiplicity decisions are a use-case definition, not an amendment to the base COR HMD. A label or datatype change is not authorised merely by selecting the View.

6 QName construction for Tavi

For each row in the HMD View, resolve module through the approved module-to-namespace registry. module denotes the governed namespace owner; the prefix is an alias, not the namespace URI. local_name is the reviewed, case-sensitive taxonomy local name. Neither name, semantic_path, nor source_bsm_id may be used to invent a replacement local_name.

Purpose	Construction	COR Journal Entry PoC example
Module namespace URI	https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/<module>	https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/cor
Tavi Class navigation anchor	<module>:p_<module>_<local_name>	cor:p_cor_entryDetail
Base Class QName (C)	<module>:<local_name>	cor:entryDetail
Semantic Attribute QName (A)	<module>:<local_name>	cor:accountNumber

The module/local_name pair is the source of qualified names. In the current candidate JSON, the p_ Class anchor is a separate binding-specific name derived deterministically from the base Class QName.

Examples:
(cor, monetaryAmount, A) → cor:monetaryAmount;
(cor, entryHeader, C) → cor:entryHeader → cor:p_cor_entryHeader.

Reference (R) occurrences do not become reported Attribute facts in the current profile and are traversed transparently for presentation; any additional R concept naming would require an approved binding rule.

The current PoC also contains names such as plt:dEntryHeader, plt:dEntryHeaderDomainClass, view:h_cor_entryHeader and view:domain_cor_entryHeader.

These use the registered plt/view resource namespaces and the selected Class identifier. Their exact construction is a binding-specific resource-naming rule, not a source HMD QName; a complete generator MUST implement one approved and collision-free naming contract for these resources.

QName validation MUST resolve every prefix, preserve letter case, reject an invalid local name or a collision, and confirm that a repeated expanded QName identifies a consistent concept within the selected Tavi taxonomy.

7 Four independent definition tables

The four column contracts below are the selected Part 5 review contracts. Their mapping to the current integer-key PoC JSON is explained in Clause 10. Source fields beyond these contracts MUST NOT be added merely to reproduce the 14-column HMD View. Formal Tavi conformance is not claimed by this review draft.

Table	Columns (in order)	JSON responsibilities
3a. Concepts and Datatype	sequence, qname, type, datatype, value_domain, semantic_path	xbrlModel.dataTypes; xbrlModel.concepts; governed Value Domain binding, if any
3b. Dimension	sequence, qname, level, type, multiplicity, semantic_path	xbrlModel.domainClasses; dimensions; domainNetworks; cubes, as allowed by the binding
3c. Presentation	sequence, qname, level, type, semantic_path	xbrlModel.networks; parent-child source/target/order
3d. Label	sequence, qname, name, label_local, definition, definition_local, semantic_path	xbrlModel.labels; ordinary and documentation labels

In every table, qname is computed from HMD View module and local_name before deriving binding-specific resource names. semantic_path remains the unchanged semantic occurrence identity used locally to establish parent and ancestor relationships; it is not JSONPath.

8 Recursive processing for the Dimension Definition Table

Rows whose HMD View type is C (Class) or R (Reference) and whose effective maximum multiplicity is one (0..1, 1, or 1..1) MUST be omitted from the Dimension Definition Table. This rule applies equally whether the original HMD already had a maximum multiplicity of one or the multiplicity was restricted to one in the HMD View. Attribute (A) rows MUST NOT be omitted by this rule, regardless of their multiplicity.

For each retained row, count all omitted C and R ancestors in the selected HMD View hierarchy and reduce its level by that count. This reduction MUST be applied recursively through nested C and R occurrences, with cumulative reductions for multiple omitted ancestors. The omission of a C or R ancestor affects all retained descendant A, C and R rows.

The HMD View itself MUST remain unchanged, including its row order, sequence, level and semantic_path.

```
derived_level = HMD_View.level – count(omitted C or R ancestors)
```

Source sequence	Type	Occurrence	View multiplicity	HMD View level	Dimension level	Occurrence
1	C	accountingEntries	1 (restricted)	1	omitted	
141	C	entryHeader	0..*	2	1	entryHeader
156	A	entryId	0..1	3	2	entryId
232	C	entryDetail	0..*	3	2	entryDetail
261	A	monetaryAmount	0..1	4	3	monetaryAmount
265	A	debitCreditIndicator	0..1	4	3	debitCreditIndicator
282	C	detailAccountIdentifier	1 (changed)	4	omitted	
283	A	accountNumber	1	5	3	accountNumber
284	A	accountDescription	0..1	5	3	accountDescription

The example is illustrative, not an approved complete HMD View. In the verified COR HMD source, detailAccountIdentifier is C (sequence 282), accountNumber is A (sequence 283), and accountDescription is A (sequence 284). Therefore, the C at 282 is omitted, while both A facts remain. The two omitted C ancestors, accountingEntries and detailAccountIdentifier, reduce accountNumber from level 5 to 3.

Dimension facts and containment are separate concerns: A rows retained in the Dimension Definition Table can populate a concept domain even when a singular containing Class was omitted. A View-root C restricted to a maximum multiplicity of one produces no occurrence dimension under the current Part 5 profile; whether and how a root-only cube without such a dimension is required must follow the pinned binding rules, not an invented default..

9 Independent table contracts

The exact column contracts are those specified in Clause 7 (3a Concepts and Datatype, 3b Dimension, 3c Presentation and 3d Label). Each table MUST be independently interpretable and MUST NOT require another derived table at generation time. The pinned Tavi profile governs the separate resource naming and other binding-specific properties not included in these columns.

Table	Proposed necessary columns	Output responsibility
Concepts and Datatype	sequence, qname, type, datatype, value_domain, semantic_path	xbrlModel.dataTypes and xbrlModel.concepts
Dimension	sequence, qname, level, type, multiplicity, semantic_path	xbrlModel.domainClasses, dimensions, domainNetworks and cubes

Presentation	sequence, qname, level, type, semantic_path	xbrlModel.networks / xbrl:parent-child relationships
Label	sequence, qname, name, label_local, definition, definition_local, semantic_path	xbrlModel.labels in English and Japanese

The original view columns that are not needed by a particular table MUST NOT be copied merely to reproduce the HMD contract. QName is computed from the namespace registered for module and the approved local_name, with any Class primary-item prefix determined by the existing Tavi binding conventions.

10 Tavi Taxonomy JSON Mapping

HMD View input	Table operation	Current PoC JSON destination
multiplicity; C/R ancestor path	Suppress exact-one C and R rows only; retain A rows including exact-one A; recursively reduce the levels of retained descendants.	xbrlModel.dimensions, domainClasses, cubes
sequence; level; semantic_path	Determine nearest displayed parent and order among siblings	xbrlModel.networks[].relationships[]: source, target, order
module; local_name; type	Construct resolved Attribute QName or Class presentation anchor	xbrlModel.concepts[].name; network targets
datatype	Resolve through approved datatype binding, not name inference	xbrlModel.dataTypes, concepts[].dataType
name; label_local	Create nonempty ordinary labels for en/ja	xbrlModel.labels: forObject, labelType, language, value
definition; definition_local	Create nonempty documentation labels for en/ja	xbrlModel.labels (documentation role)

10.1 Mapping scope, source fields and naming

This clause defines, for each of the four independent definition tables in Clause 7, which table columns determine which properties in the Tavi JSON. The only direct semantic input to each table is the selected and reviewed 14-column HMD View specified in Clause 5. Row selection and permitted multiplicity restrictions MUST be completed before table derivation. The source COR HMD for taxonomy is not re-read by a table generator.

The qname column of each definition table is constructed from the HMD View module and reviewed local_name under Clause 6. For an Attribute (A), the QName identifies the reportable concept. For a represented Class (C), its presentation anchor has the binding-specific p_<module>_<local_name> form. A Reference (R) is not a reported Attribute fact in the present profile. Determining ancestry from semantic_path does not convert that path into JSONPath.

The paths below refer to the current review-candidate object form documentInfo / xbrlModel, including xbrlModel.dataTypes, concepts, domainClasses, dimensions, domainNetworks, cubes, networks, labels and tableTemplates. They are illustrated by tavi-poc/taxonomy/tavi-tax.json (PWD 2026-09-01 candidate). A definition-table value may govern generation without being emitted as a JSON property; those cases are identified explicitly.

10.2 Concepts and Datatype Definition Table (3a)

Input columns (in order): `sequence, qname, type, datatype, value_domain, semantic_path.`

The resulting table rows MUST be sufficient to select the applicable Attribute facts and the Class anchors already determined for this View. A separate multiplicity column MUST NOT be inferred or appended to this six-column contract.

Table column	Tavi JSON property / object	Conversion rule and COR Journal Entry example
sequence	xbrlModel.dataTypes[] / concepts[] emission order	Preserve selected View order for deterministic output. The original numeric sequence is not itself written as a Tavi concept property.
qname	xbrlModel.concepts[].name; xbrlModel.dataTypes[].name	A: use the expanded concept QName, e.g. cor:accountNumber. An approved datatype-name convention creates cor:accountNumberItemType. For an eligible represented C, construct the separate p_ navigation-anchor name.
type	xbrlModel.concepts[] membership	A creates a reportable fact concept. A represented C may create a p_ anchor concept; the eligibility is resolved when the table is derived from the View. R does not automatically create a fact concept.
datatype	xbrlModel.dataTypes[].baseType; concepts[].dataType	Apply the approved semantic-datatype binding, not a naming guess. Example: String -> xs:string and concept dataType cor:accountNumberItemType; Monetary -> xbrlr:monetary for cor:monetaryAmount in the current candidate.
value_domain	Approved Value Domain binding, if present	Empty in the restricted COR example. A nonempty assignment needs an explicit, schema-supported native value-domain realisation; this six-column table does not itself prescribe an EE1 or other inferred implementation.
semantic_path	Derivation and traceability; no direct JSON property	Retain the selected occurrence identity for source verification and occurrence-specific QName resolution. Do not serialise it as JSONPath or replace the concept QName with it.

In the current COR PoC, the accountNumber fact is represented as follows. periodType and nillable in this example come from the selected Tavi binding profile and are not columns of the Concepts and Datatype Definition Table:

```
"dataTypes": [{"name": "cor:accountNumberItemType",
                        "baseType": "xs:string"}],
    "concepts": [{"name": "cor:accountNumber",
            "dataType": "cor:accountNumberItemType",
           "periodType": "duration", "nillable": true}]
```

The root Class can retain a navigation anchor without an occurrence dimension when its effective maximum is one (Clause 11.3). Consequently, the Class-anchor selection MUST be resolved as part of creating table 3a from the HMD View; the Dimension Definition Table is not its input.

10.3 Dimension Definition Table (3b)

Input columns (in order): `sequence, qname, level, type, multiplicity, semantic_path.`

This table is derived only after the selected HMD View multiplicities have been fixed. C and R rows whose effective maximum multiplicity is one (0..1, 1, or 1..1) MUST be omitted. Their retained descendants are promoted recursively, counting all omitted C and R ancestors, as required by Clause 8. A rows MUST be retained by this C/R omission rule, including Attributes with multiplicity one.

Table column	Tavi JSON property / object	Conversion rule and COR Journal Entry example
sequence	domainNetworks[].relationships[].order	Order the selected A members within each resulting Class scope by source sequence. The JSON order is the local member position, not a copy of the source sequence number.
qname	domainNetworks[].relationships[].target; names of domainClasses, dimensions and cubes	For A use the fact QName, e.g. cor:accountNumber. For a retained repeating Class, derive approved view-specific resource QNames (e.g. plt:dEntryDetail and view:h_cor_entryDetail); such names are NOT identical to the source Class QName.
level	cubes[].cubeDimensions[] scope/ancestry	Use the recursively reduced level with semantic_path to identify the containing represented Class and ordered repeating ancestors. level is a derivation input, not an emitted JSON field.
type	dimension/resource eligibility; domainNetworks members	A contributes a reportable concept-domain member. C/R share the prescribed singleton omission and recursive reduction. A retained R is not automatically granted a native typed dimension by existing Class-only dimensional clauses.
multiplicity	dimensions[] presence; cubes[].cubeDimensions[]	For a retained repeating C, a maximum greater than one requires an occurrence-key dimension in the selected View. Do not produce an occurrence dimension for a singleton C/R. The minimum alone does not establish a native mandatory-fact check.
semantic_path	domainNetworks[] grouping; cubes[] ancestor path	Identify the source occurrence, selected retained parent and repeating ancestors. Keep the original semantic path even after level reduction; do not output it as JSONPath.

The current review JSON declares the EntryDetail occurrence dimension and its typed-domain Class as separate Tavi objects. The integer key type is an profile selection requiring an explicitly recorded authority, not a value taken from the HMD View Attribute datatype:

```
"domainClasses": [{"name": "plt:dEntryDetailDomainClass",
                   "allowedDomainItem": "xs:integer"}],
"dimensions": [{"name": "plt:dEntryDetail",
                "domainClass": "plt:dEntryDetailDomainClass"}]
```

For each represented Class scope, the selected Attribute QNames determine domainNetworks[].relationships[].target. For example, the existing EntryDetail concept domain includes {"source":"xbrl:conceptDomain", "target":"cor:accountNumber", "order":2}. Its scope produces a corresponding cube such as view:h_cor_entryDetail, whose cubeDimensions include xbrl:concept with a domainNetwork and the applicable repeating ancestor dimensions. The source/target role names, concept-domain root, entity/period/unit aspects, key datatype, optional flags, cubeNetworks references and deterministic resource prefixes are fixed or reviewed binding choices, not values copied from the six table columns.

The current public integer-key Journal Entry candidate has three occurrence dimensions, including plt:dAccountingEntries. This is evidence of a three-level candidate model, not evidence that an approved 14-column restricted HMD View or the four independent tables have been generated and validated. If a separate View restricts the root to one, the root occurrence dimension is omitted in that different View as specified in Clause 12.2.

10.4 Presentation Definition Table (3c)

Input columns (in order): `sequence, qname, level, type, semantic_path`.

Presentation derivation uses only this table and the fixed Tavi binding conventions; it does not read the Dimension Definition Table. Any required transparent treatment of selected singleton Class/Reference occurrences is resolved in deriving this Presentation table from the HMD View.

Table column	Tavi JSON property / object	Conversion rule and COR Journal Entry example
sequence	xbrlModel.networks[].relationships[].order	Preserve order among immediate children of the same effective parent, assigning consecutive order values within that parent. These are not the original HMD sequence numbers.
qname	networks[].relationships[].target or source	For A, target is the Attribute QName, e.g. cor:accountNumber. For a represented C, construct the p_ navigation-anchor QName, e.g. cor:p_cor_entryDetail.
level	networks[].relationships[].source/target hierarchy	Use the Presentation table hierarchy to obtain the immediate displayed parent. A source level is not directly serialised as a relationship property.
type	Class anchor / Attribute member / transparent Reference handling	Represented C nodes become presentation anchors; A becomes a fact concept. Reference (R) is traversed transparently in the current profile and does not become a fact concept.
semantic_path	Parent resolution and duplicate-occurrence tracing	Identify the nearest presented ancestor after applicable folding while preserving the unchanged HMD View path. It is not used as a network relationship QName.

The current PoC encodes the displayed EntryDetail-to-accountNumber relationship in xbrlModel.networks[].relationships[]:

```
{"source": "cor:p_cor_entryDetail",
   "target": "cor:accountNumber",
              "order": 2}
```

The containing network uses the name view:journalEntriesHierarchy and relationshipTypeName xbrl:parent-child. These properties, and the special root source xbrl:rootSource, are determined by the approved network binding and deterministic naming conventions, rather than the five table columns alone.

10.5 Label Definition Table (3d)

Input columns (in order): `sequence, qname, name, label_local, definition, definition_local, semantic_path`.

For each nonempty source text, one language-and-role-specific label object is emitted. A Class anchor label is addressed to its p_ concept QName when that is the taxonomy object being labelled.

Table column	Tavi JSON property / object	Conversion rule and COR Journal Entry example
sequence	xbrlModel.labels[] ordering	Preserve deterministic source/role/language output order. No separate sequence property is present in the native label object.
qname	labels[].forObject	Resolve the referenced Attribute QName or represented Class anchor QName. Example: cor:accountNumber.

Table column	Tavi JSON property / object	Conversion rule and COR Journal Entry example
name	labels[].value, with labelType=xbrl:label and language=en	Use the HMD View English name as the ordinary English label; e.g. Account Number.
label_local	labels[].value, with labelType=xbrl:label and language=ja	Use the HMD View Japanese label; e.g. 主勘定科目コード.
definition	labels[].value, with labelType=xbrl:documentation and language=en	Create the English documentation label only if the supplied definition is nonempty. Do not infer missing text.
definition_local	labels[].value, with labelType=xbrl:documentation and language=ja	Create the Japanese documentation label only if the supplied local definition is nonempty.
semantic_path	Source identification only; no direct JSON property	Preserve the occurrence identity for verifying that the label belongs to the selected concept; do not serialize semantic_path as a label value.

The current PoC includes the following two ordinary labels. Documentation labels required by nonempty definitions in Clause 13.2 are separate records; they are not present in the inspected candidate for this object and MUST NOT be reported as already generated:


```
{"forObject":"cor:accountNumber", "labelType":"xbrl:label",
         "language":"en", "value":"Account Number"}
{"forObject":"cor:accountNumber", "labelType":"xbrl:label",
         "language":"ja", "value":"主勘定科目コード"}
```


10.6 JSON properties not supplied directly by the four tables

The four table contracts do not themselves supply every JSON property in a compiled Tavi taxonomy. documentInfo.documentType and namespaces, xbrlModel.name, frameworkName and version, the binding-specific p_/d_/h_/domain resource naming patterns, periodType and nillable, domainClasses[].allowedDomainItem, cubeDimensions[].optional and common xbrl:entity / xbrl:period / xbrl:unit aspects require the pinned Tavi profile, the governed namespace registry, or an approved report-binding choice. None of these values MUST be inferred from the text of a source concept name.

xbrlModel.tableTemplates is a separate report-binding component under Clause 14: physical CSV column names, occurrence-key columns, factDimensions, units and precision are supplied by the selected report mapping and the applicable Tavi profile. A Table Template MUST NOT be fabricated solely from the four definition tables or treated as a fifth semantic definition table. The four tables remain independent, although the assembled JSON objects contain QName references that MUST resolve consistently.

Appendix A identifies the currently published integer-key Journal Entry taxonomy and matched Instance module/CSV at the exact GitHub revision. Its informative tables reproduce selected verified objects and the complete 18-column data table in transposed form; the linked repository files are the unmodified complete source. Publication as an illustrative PoC does not assert derivation from a reviewed 14-column HMD View, execution of all four table-generation stages, or formal Tavi conformance.

10.7 Informative COR Journal Entry trace

HMD View selection	Definition-table derivation	Resulting Tavi JSON / reference
entryDetail: C, 0..*	3a: represented Class anchor; 3b: repeating Class dimension; 3c: presentation anchor; 3d: navigation labels	concepts[].name = cor:p_cor_entryDetail; dimensions[].name = plt:dEntryDetail; cubes[].name = view:h_cor_entryDetail
detailAccountIdentifier: C, 1	3b: omit C and promote descendants one level; 3c: apply approved folding; 3d: label retained objects only	No dDetailAccountIdentifier is generated. The accountNumber fact retains its original semantic_path.
accountNumber: A, 1	3a: resolve String type; 3b: retain as domain member despite multiplicity one; 3c: attach beneath displayed EntryDetail; 3d: en/ja labels	concepts[].name = cor:accountNumber; domainNetworks target = cor:accountNumber; networks source = cor:p_cor_entryDetail, target = cor:accountNumber

This trace illustrates the already selected COR concepts and the present candidate JSON structure; it is not evidence that a new complete HMD View, four derived tables or a formally conforming Tavi taxonomy have been generated and accepted. In particular, generation of dimensions for retained plural R occurrences is not specified by the present Class-only dimension rule in Clause 12.2; R shares the singleton omission and ancestor-level reduction rule in Clause 8, but any additional native R-dimensional resource requires an approved binding rule. The current compiled candidate also contains cor:headerEntryDate and cor:periodType, whose exact local_name origins could not be matched to the inspected accepted COR HMD. Their naming must be reconciled with the selected approved View before generation conformance can be accepted.

11 Tavi binding model

11.1 Attribute facts

Every selected effective A row MUST map to its reviewed concept qualified name and an approved native datatype. The semantic datatype and the Tavi dataType binding are separate. Date maps to the native date datatype in the current profile; a date value is YYYY-MM-DD. Date Time remains a distinct mapping. Monetary semantics MUST use the approved monetary binding rather than a name-based decimal inference. Numeric classification MUST follow the resolved datatype, not the presence of an amount-like local name.

11.2 Structural rows

C rows define semantic containment. R rows are traversed transparently for binding, while their source paths remain traceable. A non-root C with maximum one MUST be folded: it contributes no Class column or **occurrence dimension** to the **table template**, and its selected descendants attach to the nearest represented ancestor in the native hierarchy. Folding does not delete the source semantic Class or its meaning. Attributes with multiplicity 1..1 remain reportable; the omission rule applies to Class structure, not Attribute columns.

11.3 Class resources

In the current Tavi profile, a **Represented Class i**s either the root Class or a repeating Class retained as an independent structural node in the Tavi hierarchy network. Each Represented Class has a p_<module>_<local_name> navigation anchor.

A **Represented Class** contributes an **occurrence dimension** when it is repeating. The root Class retains its navigation anchor even when its maximum occurrence is restricted to one, but does not contribute an **occurrence dimension** in that case.

Cubes and concept domains are generated for represented fact scopes. A **Folded Class** has no independent navigation anchor or cube; its attributes are assigned to the nearest enclosing Represented Class scope without changing their semantic identity.

A repeating Class below a **Folded Class** remains a **Represented Class**, retaining its own navigation anchor, cube and **occurrence dimension**. Its occurrence identity is preserved within the applicable represented ancestor scope.

11.4 Native networks

The hierarchy MUST be expressed using native network relationships with stable names and ordered source/target references. This profile uses xbrl:parent-child. Every relationship property MUST be supported by the selected Tavi JSON Schema revision. Additional navigation mechanisms require a separately documented native binding.

12 Dimensional network generation

12.1 Cube fact scope

Each generated cube MUST select the applicable reportable concepts through its concept domain and the **occurrence dimension**s of their represented scope. Closure and rejection of extraneous dimensions MUST be established by the selected native JSON Schema and validator. A successful JSON load alone does not prove dimensional validation.

12.2 Occurrence dimensions

A Class contributes a typed **occurrence dimension** only when the effective view maximum exceeds one. This applies to the root as well as descendants. Thus a root restricted to 1..1 has no dimension, unlike the unconditional-root occurrence rule in Part 3. Each cube includes the repeating ancestors and its own dimension when applicable, ordered outermost to innermost. Each dimension MUST correspond one-to-one with a template occurrence-key column.

12.3 Attribute membership

Each selected Attribute MUST belong to the nearest represented ancestor scope after transparent R traversal and singular-Class folding. The concept-domain relationships and hierarchy-network relationships MUST agree. Child order is derived from source order. No Attribute may be lost merely because its parent Class has been folded.

12.4 Descendant Classes and native relationships

A repeating child Class is linked from its nearest represented ancestor. Its cube includes the required repeating-ancestor dimensions. A singular intervening Class contributes neither a dimension nor a binding node, but the review mapping MUST preserve the complete source lineage. This mapping permits reverse interpretation of facts without changing their concept identities.

12.5 Reading the dimensional hierarchy

Navigation anchors are not business facts populated by CSV cells. The network presents the projected hierarchy; concept domains select reportable Attributes; typed dimensions identify repeated occurrences. The **HMD view** supplies the full semantic lineage. These three representations MUST be interpreted together, and a network link alone MUST NOT be treated as a cardinality constraint.

12.6 Multiplicity

View restrictions determine occurrence-key generation and permissible business occurrences. Minimum occurrence bounds and required Attributes MUST be validated through explicit report or business rules when the native taxonomy cannot enforce them. Missing required values, duplicate single occurrences and excessive occurrences require dedicated tests. A cube cannot establish that a mandatory fact exists merely by defining its admissible dimensions.

12.7 Integer-key PoC and restricted-root alternative

Source or view Class	View multiplicity	Binding
accountingEntries	1..*	Root anchor, cube and dAccountingEntries in current published candidate
entryHeader	0..*	Header anchor, cube, dAccountingEntries and dEntryHeader
entryDetail	0..*	Detail anchor, cube, dAccountingEntries / dEntryHeader / dEntryDetail
detailAccountIdentifier	1	Folded under Detail; no account dimension

In the current integer-key candidate, accountNumber and accountDescription are direct presentation children of entryDetail; their original semantic paths include detailAccountIdentifier. The omitted singular Class is not an occurrence dimension. Where an alternate View contains a repeated Class, a singular intervening C/R and a repeated descendant, the two retained repeated Classes require their own occurrence dimensions; the intervening singular node contributes none. This conditional example does not remove the current candidate's separate root dimension.

13 Tavi taxonomy artefacts

13.1 Native taxonomy document

The Tavi taxonomy entry document MUST conform to the pinned Tavi JSON schema and approved project binding profile. The integer-key Journal Entry PoC in Appendix A uses documentInfo.documentType = https://xbrl.org/PWD/2026-09-01/compiled and stores the concepts, datatypes, domains, dimensions, cubes, networks, labels and Table Template under xbrlModel. This profile is a collaborative review candidate. Its JSON syntax and structural schema validation MUST NOT be represented as complete external Tavi PWD conformance.

13.2 Labels and documentation

For every generated concept or navigation anchor, nonempty name and label_local values MUST generate the applicable English and local-language ordinary labels. Nonempty definition and definition_local values MUST generate documentation labels. The current profile uses xbrl:label and xbrl:documentation with en and ja language tags. Missing source text MUST be recorded, not invented. British English prose does not require replacing en by en-GB in existing taxonomy labels.

13.3 Hierarchy network

The generator MUST emit the source-ordered projected hierarchy after non-use exclusion, R traversal and folding. Every source and target qualified name MUST resolve. Duplicate relationships with the same effective identity MUST be absent. The root is retained as a navigation anchor even when it has no **occurrence dimension**.

13.4 Network set

The package MUST identify the hierarchy networks used by its cubes. Unrelated full-model networks MUST NOT enter the restricted DTS. Additional aggregation or calculation networks require separately reviewed semantics and tests; they are not inferred from parent-child containment.

13.5 View entry point and DTS separation

The supported restricted profile generates an independent taxonomy directly from the **HMD view**. It MUST NOT import the Full Tavi model merely to filter its display. Selected import through includeObjects remains an open implementation issue and MUST NOT be relied on until exact loaded-object boundaries pass formal validation. A full taxonomy with a filtered template is a different profile and MUST NOT be labelled an independent restricted taxonomy.

13.6 Domains and cubes

Concept domains MUST contain exactly the selected Attributes for each represented scope. Cube dimensions MUST reference those domains and the applicable typed **occurrence dimension**s. Domains and cubes are view-specific structural resources. All references, memberships and ancestry MUST be checked against the approved view.

13.7 Discovery and dependency resolution

The complete loaded DTS MUST be inspected, including dependencies brought in transitively. Schema validity is insufficient to prove selected-object scope. Expected and actual concept, type, dimension, domain, cube, network and label sets MUST agree. An unresolved reference or unexpected Full-model import prevents restricted-taxonomy acceptance.

14 Instance Module and Structured Tidy Data CSV

14.1 Separation from taxonomy generation

The Tavi taxonomy defines the semantic and structural model of the selected HMD View, including concepts, datatypes, dimensions, cubes, networks and labels. It also contains the Table Template, which defines the correspondence between report aspects and physical CSV columns or constants.

A Table Template MUST NOT repair taxonomy types, rename concepts or add hierarchy by itself. A definition script MAY generate Table Templates from the reviewed HMD View and an explicit physical-column mapping; HMD semantics alone do not determine ERP-specific CSV header names.

The **Instance Module** provides the metadata required to associate a particular Structured Tidy Data CSV with the applicable Tavi taxonomy and its Table Template. It identifies the taxonomy through importMapping and importedTaxonomies, selects the Table Template through factMaps, and associates the CSV data source with the Fact Map through sourceMappings and factSources.

The **Instance Module** serves a function analogous to the JSON metadata document in xBRL-CSV. It applies the taxonomy's existing definitions to a particular CSV without redefining the taxonomy or its Table Template. Consequently, the same taxonomy and Table Template can be referenced by different Instance Modules for different CSV datasets.

In the Journal Entry PoC, these responsibilities are separated as follows:

Artefact	Responsibility
`tavi-tax.json`	Defines the Tavi taxonomy and its embedded Table Template
`tavi-ins.json`	Associates the taxonomy and Table Template with a specific Structured Tidy Data CSV

`tavi-ins.csv`	Contains the actual Journal Entry data

Note 1 — The Journal Entry PoC uses an Instance Module to provide functionality corresponding to xBRL-CSV JSON metadata. Although the Instance Module follows the JSON syntax defined by the Tavi PWD, whether this particular use of a Tavi module is permitted under the specification remains unresolved. Syntactic and structural validity alone does not establish conformance of this usage.

14.2 documentInfo

For the integer-key Journal Entry PoC, the report entry is a Tavi PWD 2026-09-01 module document with documentInfo.documentType = https://xbrl.org/PWD/2026-09-01/module. Its documentInfo.importMapping identifies ../taxonomy/tavi-tax.json, sourceMappings identifies tavi-ins.csv, and xbrlModel.importedTaxonomies, factMaps and factSources select the taxonomy model, embedded Table Template and source. All references MUST resolve relative to their actual module and taxonomy locations. The older https://xbrl.org/2021/xbrl-csv metadata form remains a separate report format and MUST NOT be represented as this current Tavi module.

14.3 Occurrence key columns

The template MUST declare columns only for the applicable repeating-Class keys. Parent keys MUST be retained on descendant rows. Singleton C rows, including a 1..1 root, contribute no occurrence-key column. A template MUST NOT generate synthetic keys for **folded Class**es or merge occurrences from different parents.

14.4 Property columns

Each reported property column MUST resolve to its selected Attribute concept. Numeric concepts MUST have an explicit, valid unit and precision policy through columns or constants as applicable. Monetary amounts require the appropriate currency unit. The implementation MUST distinguish nonnumeric values, pure numeric values and monetary values by resolved type. Account codes are identifiers and MUST retain significant leading zeroes.

14.5 One Structured CSV per report

The present integer-key example consists of one rectangular, 18-column Structured Tidy Data CSV containing seven data records: one AccountingEntries root occurrence, two EntryHeader occurrences and four EntryDetail occurrences. The selected occurrence-key columns are dAccountingEntries, dEntryHeader and dEntryDetail. Parent keys are retained on descendant records, and blank fact cells do not independently report values. This informative example does not prove that every mandatory fact or ancestor occurrence is semantically valid.

14.6 Reading a row through the cube

A populated property cell is bound to its reported Attribute concept through the selected Table Template and the applicable occurrence-key values. In the current integer-key example, root values carry dAccountingEntries; Header facts carry dAccountingEntries and dEntryHeader; Detail facts carry all three keys, including dEntryDetail. Header records with an empty Detail key MUST NOT create a spurious Detail-dimension fact. The native Tavi processor MUST resolve the complete aspect set before a populated instance can be accepted.

14.7 Report validation

Acceptance requires metadata/table loading, resolved native facts, datatype checks, unit and precision checks, dimensional and hierarchy checks, and explicit business/cardinality checks. An empty-instance PASS proves neither populated values nor requiredness. A processor exception is HOLD, even when

fact-to-CSV correspondence passes. Saved facts MUST be compared with every nonempty source cell, including their parent occurrence keys and constants.

14.8 Shared semantics and Value Domains

Shared semantic definitions retain their governed identities. Reusable definitions and dependencies are expressed as native taxonomy objects and references. value_domain assignments MUST be preserved from the approved HMD and remain independent of datatype and occurrence keys. EE1 domain/member bindings from Part 3 cannot be assumed to exist in Tavi. A nonempty Value Domain requires a documented native realisation, pinned schema support and tests; unresolved assignments MUST stop generation, without string fallback.

14.9 Implementation profile and evidence

The public integer-key Journal Entry PoC provides 20 concepts, seven data types, three domain classes, three dimensions, three concept-domain networks, three cubes, one presentation network, 20 ordinary labels and one 18-column Table Template. The source material has no xbrl:documentation labels. Published evidence covers static JSON syntax, structural schema checks, internal references, CSV/Template column agreement and static viewer display; native import resolution, fact reconstruction, requiredness and fully populated Tavi PWD validation remain unverified. This example is not an accepted substitute for the independently reviewed 14-column HMD View and four derived tables.

15 Cross binding equivalence

Equivalence is evaluated against the selected **HMD view**, not all omitted Full-HMD occurrences. Retained Attribute semantics, expanded qualified names, values and source paths MUST agree across the supported realisations. Folding may change navigation topology but MUST preserve retained business relationships and occurrence identity. Part 3 resources and Tavi resources are compared by meaning and resolved aspects, rather than physical document shape or object counts. Full-model completeness and restricted-view completeness are separate claims.

16 Accounting Entries baseline and additional HMDs

The accepted COR Accounting Entries HMD supplies the reusable semantic definitions for purpose-specific views. The current published integer-key Journal Entry PoC uses a 14-column HMD View contract in this Part but has not been demonstrated, by available public generation evidence, to have been generated from a fully reviewed 14-column View and four independent definition tables. The illustrative public taxonomy contains 20 concepts, seven data types, three occurrence dimensions, three domain classes, three concept-domain networks, three cubes, one presentation network, 20 ordinary labels and one embedded 18-column Table Template. These are observed counts for the candidate, not normative generator constants. Business Transactions and AAA remain separately governed source roots and require their own accepted views.

17 Formal package assembly

17.1 Package layout

Artefact	Purpose
view/<view-id>.csv	Reviewed 14-column **HMD view**
view/<view-id>-review.md	Source, selected paths and restriction decisions

taxonomy/<view-id>-tavi-<V>.json	Proposed view-derived taxonomy; Appendix A uses the existing tavi-poc/taxonomy/tavi-tax.json candidate
instance/<report>.json and <report>.csv	Proposed matched module and CSV; current public example is tavi-poc/instance/tavi-ins.json and tavi-ins.csv
tools and generator specification	Approved definition interfaces and implementation
validation report	Separate taxonomy, template and populated-report results

17.2 Version and namespaces

V MUST be an explicit release date in YYYY-MM-DD form. Candidate status MAY be recorded separately. Project semantic module namespaces remain https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/<module>. View-resource identities MUST distinguish view-specific constraints; a date in a physical filename MUST NOT silently change a semantic module namespace. External namespace identifiers follow the pinned native profile.

17.3 Deterministic view identifier

A view identifier MUST be stable and unique within the package, associated with one source root and reviewed purpose. Identifiers and naming rules MUST be documented. Deterministic generation MUST preserve ordered input relationships and produce equivalent resolved objects for the same accepted inputs and bindings.

17.4 Publication boundary

A **view taxonomy** MAY be accepted separately from its report package when clearly labelled. Complete report-package publication requires all applicable taxonomy, template and populated-report tests to pass. HOLD artefacts and earlier evidence MUST be preserved. Candidate acceptance does not itself authorise Canonical promotion or publication in Git.

18 Validation and conformance

18.1 Taxonomy conformance

A conformant result MUST pass source/view identity and interval checks, non-use exclusion, qualified name/datatype resolution, JSON Schema validation, complete native DTS inspection, exact dependency scope, domain/cube/network correspondence and label/documentation comparison. Single Classes MUST contribute no **occurrence dimensions**. Taxonomy validation MUST finish successfully before dependent template validation proceeds.

18.2 Minimum report tests

Tests MUST cover repeated Header/Detail occurrences; repeating roots and roots restricted to one; repeat–singular–repeat nesting; Reference traversal; folded account code/name placement; leading-zero identifiers; Date versus Date Time; monetary and pure precision policies; absent, nil and zero values; parent-key preservation; missing required values and excess singleton occurrences; and rejection of undeclared aspects where closure applies. Existing valid evidence may be reused for unchanged conditions. At least one populated fixture MUST be formally validated and reconciled to its CSV. An exception or unsupported cardinality check MUST be reported as unresolved rather than passed.

19 Output name summary

Output	Rule
Formal source HMD	Accepted 18-column HMD for taxonomy (source). Separate derived HMD View is 14 columns.
View CSV	<view-id>.csv; exact 14-column contract
Tavi taxonomy	Proposed pattern <view-id>-tavi-<V>.json; current published candidate tavi-poc/taxonomy/tavi-tax.json
Report pair	For current integer-key candidate: tavi-poc/instance/tavi-ins.json plus tavi-ins.csv
Review record	<view-id>-review.md; separate from semantic CSV

The file patterns above are proposed release naming conventions, not alternative names for the public review candidate. Appendix A refers to the existing published filenames and does not rename or replace them. Review of any new release must distinguish its accepted 14-column View and four generated tables from the already published example.

20 Instance examples

20.1 Example files

The current published integer-key Journal Entry example consists of taxonomy/tavi-tax.json, instance/tavi-ins.json and instance/tavi-ins.csv under tavi-poc/, as identified in Appendix A. Its taxonomy contains one embedded 18-column Table Template, and its CSV has seven occurrence records. It is distinct from the withdrawn nine-column tavi-inst.csv and 2021 xBRL-CSV metadata examples. The official published source bytes, not the informative transposed display in this document, determine the exact report values and column order.

20.2 Accounting Entries example

CSV column	Concept or aspect
dAccountingEntries	plt:dAccountingEntries
dEntryHeader	plt:dEntryHeader
dEntryDetail	plt:dEntryDetail
documentInformationDocumentType	cor:documentInformationDocumentType
auditNumber	cor:auditNumber
defaultCurrency	cor:defaultCurrency
periodCoveredStart	cor:periodCoveredStart
periodCoveredEnd	cor:periodCoveredEnd
periodType	cor:periodType
entityPartyId	cor:entityPartyId
entityPartyName	cor:entityPartyName
entryNumber	cor:entryId
entryDate	cor:headerEntryDate
drCrIndicator	cor:debitCreditIndicator

accountCode	cor:accountNumber
accountName	cor:accountDescription
amount	cor:monetaryAmount
description	cor:entryDetailDescription

The integer-key example has one AccountingEntries root record with reporting defaults and entity information, two EntryHeader records for JE001 and JE002 on 2026-10-10, and four EntryDetail records. JE001 reports Dr 100 Cash / Cr 400 Sales, each JPY 1,000. JE002 reports Dr 120 AccountsReceivable / Cr 400 Sales, each JPY 2,000. Each of the four Detail records contains debit/credit indicator, account code/name, amount and description. The example contains 18 physical CSV columns, including three occurrence keys, and does not substantiate the older nine-column, 24-fact validation claim.

20.3 Additional views

Business Transactions and AAA examples require their own approved selections, multiplicity restrictions and source-column mappings. They MUST NOT reuse Accounting Entries metadata merely by replacing the CSV URL. A metadata/data pair belongs to the report it actually describes.

20.4 Concepts dimensions and presentation

The public integer-key taxonomy retains three occurrence dimensions: plt:dAccountingEntries, plt:dEntryHeader and plt:dEntryDetail. Its EntryDetail presentation network links cor:accountNumber and cor:accountDescription directly beneath cor:p_cor_entryDetail, reflecting the omission of the singular detailAccountIdentifier container in that projected view. This does not alter either Attribute's source semantic_path. A different HMD View with root multiplicity restricted to one would omit the root occurrence dimension and is not the candidate shown in Appendix A.

20.5 Unit handling and validation status

The current integer-key CSV reports iso4217:JPY in its defaultCurrency property, with amounts 1000 or 2000 and date-only entryDate values of 2026-10-10. Monetary-unit and decimals aspects must be resolved through the embedded Table Template and applicable report binding; the physical CSV does not by itself validate them. Earlier October 12 B1–B5 results belong to a withdrawn nine-column experiment and MUST NOT be reported as current populated-instance validation. For the November 2 review candidate, JSON syntax, external structural checks, CSV/Table Template column consistency and static viewer display were recorded, while native import, fact reconstruction and full Tavi PWD semantic conformance remain unresolved.

Appendix A
(informative)
Tavi taxonomy

A.1 Public integer-key Journal Entry taxonomy: tavi-tax.json

This informative example refers to the integer occurrence-key candidate published in the XBRL GL Next GitHub repository at revision a9d96d92925d9f8e5610b865f5e46cde24a2221d (2 November 2026). It replaces the historical nine-column documentInfo/taxonomy example. The linked GitHub JSON is the complete source; the following tables describe selected verified properties without modifying the source file.

Published complete source (immutable revision):

https://github.com/pontsoleil/XBRL_GL_Next/blob/main/tavi-poc/taxonomy/tavi-tax.json

Property	Verified value / role
documentInfo.documentType	https://xbrl.org/PWD/2026-09-01/compiled
xbrlModel.name	view:accountingEntriesJournalPoC
xbrlModel.version	2026-11-01-18-column-three-level-candidate
Namespaces	COR and other governed module namespaces; plt for occurrence resources; view for the PoC
Occurrence key type	xs:integer in domainClasses[].allowedDomainItem
Table Template	view:accountingEntriesJournalPoC18Columns; 18 columns embedded in xbrlModel.tableTemplates

xbrlModel array	Objects	Example
dataTypes	7	cor:accountNumberItemType (baseType xs:string)
concepts	20	cor:accountNumber; cor:p_cor_entryDetail
domainClasses	3	plt:dEntryDetailDomainClass
dimensions	3	plt:dAccountingEntries; plt:dEntryHeader; plt:dEntryDetail
domainNetworks	3	view:domain_cor_entryDetail
cubes	3	view:h_cor_entryDetail
networks	1	view:journalEntriesHierarchy (xbrl:parent-child)
labels	20	Ordinary en/ja labels, no documentation labels in this candidate
tableTemplates	1	view:accountingEntriesJournalPoC18Columns

Illustrative native JSON properties from the published taxonomy (not the full source listing):

```
"dataTypes": [
  {"name":"cor:accountNumberItemType", "baseType":"xs:string"},
  . . .
],
"concepts": [
  {
    "name":"cor:accountNumber",
```

```
    "dataType":"cor:accountNumberItemType",
    "periodType":"duration",
    "nillable":true
  },
  . . .
],
"domainClasses": [
  {"name":"plt:dEntryDetailDomainClass", "allowedDomainItem":"xs:integer"},
  . . .
],
"dimensions": [
  {"name":"plt:dEntryDetail", "domainClass":"plt:dEntryDetailDomainClass"},
    . . .
],
"networks": [
  {
    "name":"view:journalEntriesHierarchy",
    "relationshipTypeName":"xbrl:parent-child",
    "relationships":[...]
  }
]
```

The published integer-key taxonomy has three dimensions, three cubes and one 18-column Table Template, and it is not the same physical example as a separate HMD View with a singleton root and no root occurrence dimension. The repository publishes this JSON as an experimental collaborative review candidate, not as conclusive evidence of full Part 5 or Tavi PWD conformance.

A.2 Public integer-key Journal Entry instance module: tavi-ins.json

This is a Tavi PWD 2026-09-01 module with documentInfo.importMapping, xbrlModel.importedTaxonomies, factMaps and factSources. It is not the withdrawn 2021 xBRL-CSV tableTemplates document shown in the earlier Appendix A.

Published complete source (immutable revision):
https://github.com/pontsoleil/XBRL_GL_Next/blob/a9d96d92925d9f8e5610b865f5e46cde24a2221d/tavi-poc/instance/tavi-ins.json

JSON property	Published value
documentInfo.documentType	https://xbrl.org/PWD/2026-09-01/module
documentInfo.importMapping key	view:accountingEntriesJournalPoC
documentInfo.importMapping path	../taxonomy/tavi-tax.json
xbrlModel.name	view:accountingEntriesJournalCSVInstance
xbrlModel.importedTaxonomies[0].xbrlModelName	view:accountingEntriesJournalPoC
xbrlModel.factMaps[0].templateName	view:accountingEntriesJournalPoC18Columns
xbrlModel.factSources[0].factMapName	view:accountingEntriesJournalCSVMap
documentInfo.sourceMappings[0].url	tavi-ins.csv

The metadata cross-references are internally consistent in the inspected published files. This is a static correspondence check, not evidence of native Tavi import resolution or fact reconstruction.

A.3 Public integer-key Structured Tidy Data: tavi-ins.csv

The current physical instance CSV has 18 columns and seven occurrence records (plus one header). The three key columns are dAccountingEntries, dEntryHeader and dEntryDetail. The table below is

transposed for readability; this transposition does not change the published CSV or its physical column order.

Published complete source (immutable revision):
https://github.com/pontsoleil/XBRL_GL_Next/blob/a9d96d92925d9f8e5610b865f5e46cde24a2221d/tavi-poc/instance/tavi-ins.csv

Field	Record 1	Record 2	Record 3	Record 4	Record 5	Record 6	Record 7
dAccountingEntries	1	1	1	1	1	1	1
dEntryHeader	—	1	1	1	2	2	2
dEntryDetail	—	—	1	2	—	1	2
documentInformation DocumentType	—	—	—	—	—	—	—
auditNumber	—	—	—	—	—	—	—
defaultCurrency	iso4217:JPY	—	—	—	—	—	—
periodCoveredStart	2026-04-01	—	—	—	—	—	—
periodCoveredEnd	2027-03-31	—	—	—	—	—	—
periodType	duration	—	—	—	—	—	—
entityPartyId	ABC123	—	—	—	—	—	—
entityPartyName	Example Co.	—	—	—	—	—	—
entryNumber	—	JE001	—	—	JE002	—	—
entryDate	—	2026-10-10	—	—	2026-10-10	—	—
drCrIndicator	—	—	Dr	Cr	—	Dr	Cr
accountCode	—	—	100	400	—	120	400
accountName	—	—	Cash	Sales	—	Accounts Receivable	Sales
amount	—	—	1000	1000	—	2000	2000
description	—	—	Receipt	Sale	—	Invoice	Sale

Provenance: published at GitHub with the matched JSON taxonomy, Tavi module and 18-column CSV. Existing review checks do not establish that an accepted 14-column HMD View, four independent definition tables or a current generation script produced these exact source files. No fresh native Tavi or populated-instance validation is claimed in this revision.

Appendix B
References

(Reference list under collaborative review. Official XBRL specifications are available at https://specifications.xbrl.org/.)

RFC2119	The Internet Engineering Task Force (IETF). "Key words for use in RFCs to Indicate Requirement Levels, March 1997" Scott Bradner. (See http://www.ietf.org/rfc/rfc2119.txt)
RFC9535	Internet Engineering Task Force (IETF). “JSONPath: Query Expressions for JSON”, RFC 9535, February 2024 Stefan Gössner, Glyn Normington, Carsten Bormann (See https://www.rfc-editor.org/rfc/rfc9535.html)
TRP	XBRL International Inc. (XII). "XBRL Taxonomy Recognition Process, 28 October 2004" Peter Calvert, and Joseph MacDonald. (See https://xbrl.org/taxonomyrecognition/taxonomy-recognition-acknowledgement-process-2013-04-09.htm)
XBRL 2.1	XBRL International Inc. (XII). "Extensible Business Reporting Language (XBRL) 2.1 Specification with corrected errata dated 2 July 2008" Phillip Engel, Walter Hamscher, Geoff Shuetrim, David vun Kannon, and Hugh Wallis. (See https://specifications.xbrl.org/)
GXLTFTA	XBRL International Inc. (XII). “XBRL GL Taxonomy Framework Technical Architecture 2015” Gianluca Garbellotto, Eric E. Cohen, Masatomo Goto, Toshimitsu Suzuki (See https://www.xbrl.org/int/gl/2015-03-25/GLTFTA-REC-2015-03-25.html)
XBRL GL Framework	XBRL International Inc. (XII). “XBRL Global Ledger Taxonomy Framework 2015“ Eric E. Cohen, Masatomo Goto, Toshimitsu Suzuki (See https://www.xbrl.org/int/gl/2015-03-25/gl-framework-REC-2015-03-25.html)
XBRL Tavi	XBRL International Inc. (XII). “Project Tavi 1.0 Public Working Draft 1 September 2026” Herm Fischer, Campbell Pryde (see https://www.xbrl.org/Specification/tavi/PWD-2026-09-01/tavi-PWD-2026-09-01.html)
XBRL GL Next Requirements Specification	supplied collaborative draft
XBRL GL Next Taxonomy Framework Parts 1–4	supplied drafts, including Part 3 dated 15 September 2026

Appendix C
Intellectual property status

Copyright and license terms are under review. This draft is made available through the public project repository for collaborative review. Publication of this draft does not establish additional redistribution or derivative-use rights.

References to external specifications and other third-party materials remain subject to the terms of their respective rights holders.

This draft does not claim approval, endorsement, publication, or official status by XBRL Japan, XBRL Europe, XBRL International, or any other organization.

Appendix D
Document history

Date	Details
12 October 2026	Initial collaborative review draft.
3 November 2026	Review revision based on the published integer-key Journal Entry PoC. Clarified the 14-column HMD View, four independent definition tables, C/R singleton reduction and current PWD 2026-09-01 documentInfo/xbrlModel and module/CSV structures. Replaced the withdrawn nine-column Appendix A example with an informative summary and transposed data from the current integer-key taxonomy, Instance module and CSV.
6 November 2026	Revised the correspondence with the XBRL GL Next Requirements Specification to reflect the updated requirements.

Appendix E
(informative)
Correspondence with XBRL GL Next Requirements

Table E.1 maps the applicable requirements of the XBRL GL Next Requirements Specification (27 October 2026 revision) to their implementation in this Part. This informative correspondence does not create new GL-REQ identifiers or assert separate conformance.

Table E.1 — Mapping of XBRL GL Next Requirements

Requirement ID	Requirement Name	Implementation in Part 5
GL-REQ-0001	Consistent business semantics across representations	Derives the Tavi-specific 14-column semantic projection from an accepted HMD for taxonomy, preserves semantic paths and reviewed meanings, and separates HMD-view selection from native taxonomy and table-template generation (Clauses 4.1–4.4).
GL-REQ-0107	Business completeness and data quality	Defines source/view identity checks, datatype and QName resolution, native DTS inspection, dimensional constraints, report reconciliation and explicit cardinality checks; an unresolved populated-instance test remains HOLD rather than a conformance claim (Clauses 4, 8.7 and 12).
GL-REQ-0301	Exchange-format independence	Uses an approved Accounting Entries HMD as the semantic source for a proposed purpose-specific Tavi View; other domain views require their own approved selections (Clauses 4–5 and 16).
GL-REQ-0201	Reusable business scope	Defines purpose-specific, independently generated native JSON view taxonomies, their data types, concepts, dimensions, domains, cubes, networks and view-specific entry points, with deterministic package naming (Clauses 7 and 11).
GL-REQ-0201	Reusable business scope	Binds each purpose-specific HMD view to its selected source root, reviewed scope and distinct taxonomy entry point rather than merging different message roots or assuming that a full-model taxonomy is a restricted view (Clauses 4.6, 7.5 and 11.3).
GL-REQ-0301	Exchange-format independence	Provides a native JSON taxonomy realisation alongside the XBRL 2.1 Tuple and OIM realisations, preserving the selected semantic meaning while keeping physical representations and their validation boundaries separate (Clauses 2.1 and 9).
GL-REQ-0301	Exchange-format independence	Specifies Tavi data types, concepts, native hierarchy relationships, domains, cubes and dimensions, and binds the selected taxonomy to xBRL-CSV metadata and a corresponding Structured CSV report (Clauses 5–8).
GL-REQ-0303	Identity and relationships of repeated business information	Retains source semantic_path, concept identity and source Class lineage through view projection and singleton-Class folding; cross-binding equivalence compares retained

Requirement ID	Requirement Name	Implementation in Part 5
		meanings and resolved aspects rather than document shape (Clauses 4.3, 5.2 and 9).
GL-REQ-0303	Identity and relationships of repeated business information	Creates occurrence dimensions only for repeating Classes in the selected view, folds singular Classes without discarding semantics, and maps occurrence-key columns to applicable dimensions and cubes (Clauses 5.3, 6 and 8.3).
GL-REQ-0401	Accessible examples and independent inspection	Provides the current public integer-key Journal Entry candidate as an informative example: an 18-column Structured Tidy Data CSV, its matching Instance Module and Tavi taxonomy. The example does not establish formal populated-instance or full Tavi conformance (Clauses 14, 20 and Appendix A).
GL-REQ-0402	Demonstrable business interoperability	Requires reproducible source/view reconciliation, schema and loaded-DTS checks, template and populated-report validation, and reporting of unresolved processor or cardinality conditions; past PASS and B5 HOLD evidence remain distinguished (Clauses 8.7, 8.9, 11.4 and 12).