XBRL GL Next Taxonomy Framework — Part 3: OIM 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.

Abstract

This Part specifies the OIM-compatible palette taxonomy and xBRL-CSV report binding of XBRL GL Next, using the same formal HMD-for-taxonomy input set as the Tuple realisation in Part 2. It maps Attribute occurrences to reportable concepts and preserves Class hierarchy through primary-item anchors, closed hypercubes, dimensional relationships and targetRole navigation.

Occurrence-key typed dimensions are generated for the root and repeated Classes under the Part 1 rule; singular Classes remain part of the hierarchy without unnecessary occurrence dimensions. This Part also defines root-specific OIM entry points, module schemas, presentation and definition networks, Value Domain bindings, Structured CSV occurrence columns, JSON report metadata and the corresponding validation conditions.

Accounting Entries and Business Transactions remain independent root-specific DTSs. Cross-binding correspondence is assessed by business meaning, semantic paths, fact identity, occurrence context and relationships rather than by requiring OIM dimensional structures to reproduce XML Tuple nesting.

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 Authoritative input and validation contract 6
4.1 Input authority 6
4.2 Required 18-column order 6
4.3 XPath and QName authority 6
4.4 Row validation 7
4.5 Non-use subtrees 7
4.6 Cross-HMD shared declarations 7
5 OIM binding model 8
5.1 Attribute facts 8
5.2 Structural rows 8
5.3 Class resources 8
5.4 Extended roles 8
6 Dimensional network generation 8
6.1 Closed cube 8
6.2 Occurrence dimensions 9
6.3 Attribute membership 9
6.4 Descendant Classes and targetRole 9
6.5 Reading the dimensional hierarchy 9
6.6 Multiplicity 9
6.7 Occurrence-key alignment example 10
7 OIM taxonomy artefacts 10
7.1 OIM module schema 10
7.2 OIM labels 10
7.3 OIM module presentation linkbase 10
7.4 OIM Presentation Linkbase Set 10
7.5 OIM message entry point and DTS separation 11
7.6 Dimensional definition linkbase 11
7.7 Linkroles and DTS discovery 11
8 xBRL-CSV metadata and data tables 11
8.1 Separation from taxonomy generation 11
8.2 documentInfo 11
8.3 Occurrence-key columns 11
8.4 Property columns 12
8.5 One Structured CSV per report 12
8.6 Reading a Structured CSV row through the cube 12
8.7 Report validation 12
8.8 Shared `gen` module, Value Domains, and Structured CSV 12
8.9 Implemented occurrence and Value Domain rules 13
9 Cross-binding equivalence 13

10 accountingEntries baseline and businessTransactions additional HMD (informative)... 14
11 Formal package assembly ... 14
11.1 Formal package tree ... 14
11.2 Version and namespaces ... 15
11.3 Deterministic HMD identifier ... 15
11.4 All-or-nothing publication ... 15
12 Validation and conformance ... 15
12.1 Taxonomy conformance ... 15
12.2 Minimum report tests ... 16
13 Normative output-name summary ... 16
14 xBRL-CSV instance examples (informative) ... 16
14.1 Example files ... 16
14.2 Accounting Entries example ... 17
14.3 Business Transactions example ... 20
14.4 Concepts, dimensions, and presentation ... 24
14.5 Unit handling ... 24
Appendix A References ... 25
Appendix B Intellectual property status ... 26
Appendix C Document history ... 27
Appendix D Errata corrections in this document ... 29
Appendix E (informative) Correspondence with XBRL GL Next Requirements ... 30

1 Language independence

This collaborative review draft is written in English. The preferred spelling convention is UK English. XML names, QNames, namespace URIs, filenames, command-line options, and diagnostic identifiers are case-sensitive and are not translated.

2 Introduction

2.1 Scope

This Part defines the OIM-compatible taxonomy binding and xBRL-CSV report-metadata rules for XBRL GL Next. It distinguishes taxonomy generation from report generation and defines an OIM module palette that is independent of the Tuple module palette. It does not translate XML tuples into OIM facts; Tuple and OIM realisations are generated independently from the same **HMD for taxonomy** input set.

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 1 defines the serialisation-independent semantic pipeline and the formal HMD contract. Part 2 defines the XBRL 2.1 Tuple realisation generated from formal HMDs. This Part 3 defines the OIM realisation generated from the same HMDs. Part 4 governs extension classification, registration, alignment and publication. Part 5 defines purpose-specific HMD views and their Tavi taxonomy and Table Template realisation.

This Part 3 specifies how formal HMDs are realised as OIM taxonomies, including primary items, dimensions, hypercubes, occurrence identification, and the corresponding taxonomy generation requirements.

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. Each HMD serves as a purpose-specific palette, selecting reusable semantics from COR, BTX and AAA, which constitute the semantic paint boxes.

2.4 Implementation independence

The requirements in this Part define observable inputs, validation outcomes, generated taxonomy components, and deterministic relationships among them. The sole maintained reference Generator authority is `tools/taxonomy/xBRLGL_TaxonomyGenerator.py`. The Generator and its supporting modules provide datatype mapping/override, EE1 Value Domain, selector-effective multiplicity, occurrence-dimension, Tuple/OIM, manifest, and deterministic-generation processing. `tools/taxonomy/datatype_binding.py` is a required runtime dependency;

3 Terms and definitions

For the purposes of this document, the terms and definitions given in **Part 1** 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.

HMD root
the effective level-1 root Class of one **HMD for taxonomy**. In the semantic model it is identified by its Class semantic-model identifier; the selected syntax binding assigns the applicable QName.

formal HMD
one HMD_for_taxonomy CSV produced by Post-Graph Walk and containing exactly one effective level-1 root.

HMD identifier
the stable filesystem identifier obtained from the **HMD root** module identifier, an underscore, and the root local name; for example cor_accountingEntries.

OIM module schema
the standalone-valid <module>-oim-<V>.xsd schema that owns OIM Attribute item declarations and module-owned Class primary-item anchors p_<module>_<local_name>. It contains no Tuple C/R declarations or Tuple structural complexTypes.

OIM module presentation linkbase
the <module>-oim-pre-<V>.xml presentation linkbase for one module. Class occurrences are represented by p_ primary items, Attributes by their OIM item concepts, and Reference rows are traversed transparently.

OIM Presentation Linkbase Set
the OIM binding-specific set of module OIM presentation linkbases published with the taxonomy package. An **OIM message entry point** discovers the member corresponding to its **HMD root** module.

OIM message entry point
the root-specific <P>-all-oim-<V>.xsd schema through which one HMD OIM DTS is discovered.

primary item
an item concept used as an anchor for a Class scope. Module-level p_ Class **primary item**s participate in OIM presentation and dimensional primary-item networks but are not themselves the Attribute facts reported by the example xBRL-CSV instances.

hypercube
a dimensional constraint identifying the dimensions applicable to a set of **primary items**; each effective Class scope has one closed **hypercube** in the selected HMD OIM DTS.

occurrence dimension
a generated typed dimension whose value identifies one occurrence of an **occurrence-key Class** within an HMD hierarchy. It corresponds one-to-one with the applicable **Structured CSV** occurrence-key column.

occurrence-key Class
a Class classified under Part 1 as requiring explicit occurrence identity: the root Class or a Class whose effective maximum occurrence is greater than one.

targetRole
the xbrldt:targetRole attribute on a dimensional relationship that directs continued traversal in another extended link role.

formal package
the complete generated package containing separate Tuple and OIM module palettes and one root-specific Tuple and OIM message entry point for every formal HMD in the accepted **HMD for taxonomy** input set.

`gen` taxonomy module
the one shared generated taxonomy module containing binding-neutral reusable semantic definitions and used by both Tuple and OIM taxonomy assemblies.

enumeration Value Domain
a **Value Domain** whose approved XBRL realisation constrains a fact to one member of a governed permitted value set.

4 Authoritative input and validation contract

4.1 Input authority

A conformant taxonomy generator MUST accept the HMD for taxonomy input set defined in Part 1. The set contains one or more formal 18-column HMD_for_taxonomy files generated from approved LHM-reviewed inputs. Each HMD is authoritative for occurrence hierarchy, semantic-path correspondence, Attribute QNames, semantic datatypes, reviewed multiplicities, reviewed `value_domain` assignments, labels, definitions, and the Class structure from which the OIM dimensional binding is generated. XBRL item types and **Value Domain** realisations are explicit taxonomy bindings driven by those authoritative inputs.

Each **formal HMD** MUST contain exactly one effective level-1 root. Every HMD in the set MUST satisfy Clause 4 before the **formal package** is published. Implementation-specific manifests and execution records are outside the normative input contract.

4.2 Required 18-column order

```
sequence,module,level,type,identifier,name,datatype,multiplicity,value_domain,definit
ion,label_local,definition_local,source_bsm_id,semantic_path,associated_module,class_
term,local_name,xpath
```

The header names and order MUST match exactly. A byte-order mark MAY precede the first header name. Empty data rows MAY be ignored. No additional column is permitted in a conforming **formal HMD**.

4.3 XPath and QName authority

The xpath column is authoritative for the physical hierarchy. Every XPath MUST start with /xbrli:xbrl/ and every subsequent step MUST be a lexical QName of the form gl-<module>:<local-name>. The number of QName steps after xbrli:xbrl MUST equal level.

For each row, the terminal XPath step MUST resolve to the namespace for module and MUST have the local part given by local_name. The generator MUST NOT reconstruct the XPath from level, module, or local_name. It MUST reject a mismatch.

HMD	Root module	Root local_name	Normative root XPath	HMD identifier
Accounting Entries	cor	accountingEntries	/xbrli:xbrl/gl-cor:accountingEntries	cor_accountingEntries
Business Transactions	btx	businessTransactions	/xbrli:xbrl/gl-btx:businessTransactions	btx_businessTransactions

4.4 Row validation

- sequence MUST be an integer and input order MUST determine the order of direct children within a content model.
- level MUST be a positive integer and MUST equal the number of physical QName steps in xpath.
- type MUST be C, A, or R. C denotes Class, A denotes Attribute, and R denotes Reference or other structural Association occurrence.
- module and local_name MUST each be a valid XML Schema NCName. local_name MUST NOT contain a colon.
- multiplicity MUST be one of 0..0, 1, 1..1, 0..1, 0..*, or 1..*.
- C and R rows MUST NOT specify a datatype. Every A row MUST specify a semantic datatype that resolves to an XBRL item type through an approved explicit override or default mapping; an unknown or `review-required` result is an error.
- semantic_path MUST be unique within the HMD.
- Every non-root physical parent XPath MUST precede its child in the same HMD.
- The same expanded QName MUST NOT have conflicting row type or datatype definitions.

4.5 Non-use subtrees

A row with multiplicity 0..0 and every physical descendant of that row MUST be excluded from generated taxonomy content. The generator MAY verify such rows, but MUST NOT emit declarations, relationships, or entry-point dependencies for the excluded subtree. This rule is a defensive validation rule; normally Post-Graph Walk has already removed the non-use subtree from the **formal HMD**.

4.6 Cross-HMD shared declarations

The conforming result MUST be independent of HMD input order. Cross-HMD equality requirements apply only to declarations deliberately governed as shared. A deliberately shared Attribute QName MUST have identical concept-defining values, and a deliberately shared Class primary-item QName MUST resolve to the same module-owned p_ identity. HMD-specific hypercubes, **occurrence dimensions**, closed-cube ancestry, and dimensional relationships are root-specific and MAY differ between independent OIM DTSs.

Within one OIM DTS, every required declaration and dimensional relationship MUST resolve consistently. A conflict in a deliberately shared declaration is an error and MUST be resolved before publication. Mere reuse of a QName in another independent root-specific DTS does not by itself create a cross-DTS equality requirement.

5 OIM binding model

5.1 Attribute facts

Every effective A row MUST be bound to one XBRL item concept in the **OIM module schema** of its owning module. The fact QName MUST equal the expanded QName resolved from the terminal HMD XPath step. The **OIM module schema** MUST be standalone-valid and MUST NOT depend on Tuple content schemas.

5.2 Structural rows

A C row establishes a semantic Class scope. The OIM module schema of the owning module MUST declare one p_<module>_<local_name> primary-item anchor for every effective Class, irrespective of occurrence-key classification. Only an **occurrence-key Class** receives an HMD-specific typed **occurrence dimension** d_Q. An R row does not create an OIM fact or module-level primary item; it is traversed transparently when assigning descendant Attributes and Classes.

5.3 Class resources

For each effective Class implementation id Q = <module>_<local_name>, the OIM binding uses the following resources.

Resource	Name	Location and role
Primary item	p_Q	owning <module>-oim-<V>.xsd; reusable module-level Class anchor
Hypercube	h_Q	selected HMD OIM entry-point schema; closed cube for the Class scope
Typed dimension	d_Q	selected HMD OIM entry-point schema; identifies an **occurrence-key Class** occurrence; emitted only when Q is an **occurrence-key Class**
Typed-domain element	_v	selected HMD OIM entry-point schema; string-valued typed-domain target; emitted only with the applicable occurrence-key dimension

The module-level p_Q anchor is binding-specific to OIM presentation and primary-item networks. For every effective Class, the OIM binding MUST retain its p_Q anchor and h_Q hypercube, a Class role, and the hierarchy representation and targetRole relationships required to preserve semantic scope. The d_Q **occurrence dimension** and its typed-domain element are conditional resources and MUST be generated only for an **occurrence-key Class** because dimensional constraints depend on the selected root hierarchy, multiplicity, and occurrence-key ancestor sequence.

5.4 Extended roles

Each effective Class Q MUST have one extended role with id link_Q and role URI http://www.xbrl.org/xbrl-gl/role/link_Q. The roleType declaration MUST be discoverable from the selected HMD OIM entry point and MUST be used by the definition link describing that Class occurrence scope.

6 Dimensional network generation

6.1 Closed cube

Within the Class role, the module-level **primary item** p_Q MUST have an all relationship to the HMD-specific **hypercube** h_Q. The all arc MUST have xbrldt:closed=true and xbrldt:contextElement=segment.

6.2 Occurrence dimensions

For each Class hypercube, the hypercube MUST have one hypercube-dimension relationship to each **occurrence-key Class** dimension on the applicable ancestor path, plus the current Class dimension when the current Class is itself an **occurrence-key Class**. Non-occurrence Class ancestors MUST NOT contribute a typed dimension. Dimensions MUST be ordered from outermost to innermost occurrence scope, and each MUST correspond to the applicable **Structured CSV** occurrence-key column.

6.3 Attribute membership

Every Attribute belonging to the current Class scope MUST be connected from p_Q by a domain-member relationship in HMD order. The Attribute locator MUST resolve to the item declaration in the owning <module>-oim-<V>.xsd. An R row MUST not interrupt the current Class scope; its descendant Attributes remain members of the current cube unless a descendant C row starts a new Class role.

6.4 Descendant Classes and targetRole

When traversal reaches a descendant Class Q2, p_Q MUST have a domain-member relationship to the module-level p_Q2 with xbrldt:**targetRole** equal to the role URI for link_Q2. The target role MUST contain the closed cube for Q2. This preserves HMD nesting without treating C or R rows as reported facts.

6.5 Reading the dimensional hierarchy

A Class is a semantic scope rather than a reported fact. Its module-level p_ anchor identifies the Class in presentation and primary-item networks, and its HMD-specific h_ concept may constrain facts for the selected root. A d_ concept exists only when the Class is an **occurrence-key Class**. Attribute items are the reportable concepts. Child Class facts retain only the required occurrence-key ancestor dimensions, while targetRole connects all applicable Class networks, including singular Classes without d_Q.

Relationship	Meaning	Effect on xBRL-CSV
p_Q —all→ h_Q	closed cube applies to facts in the Class primary-item network	only dimensions admitted by h_Q may occur
h_Q —hypercube-dimension→ d_Q	d_Q identifies a Class occurrence	occurrence-key column supplies the typed value
p_Q —domain-member→ Attribute	Attribute item is reportable within the Class scope	property column creates an Attribute fact
p_Q —domain-member/targetRole→ p_Q2	Q2 is a descendant Class with its own role	child facts carry the applicable occurrence-key ancestor dimensions and, when Q2 is an **occurrence-key Class**, the Q2 occurrence key

6.6 Multiplicity

Occurrence-key classification MUST use the Part 1 rule and the same authoritative effective HMD multiplicity used by the **Structured CSV** generator. The root Class and every Class whose maximum occurrence is greater than one require occurrence identity; a non-root `0..1` or `1..1` Class does not. A repeating Class occurrence MUST be distinguishable by its typed dimension value. Mandatory lower bounds require validation or report-generation rules because dimensional constraints alone do not guarantee that at least one fact exists for each semantic occurrence.

6.7 Occurrence-key alignment example

```
Invoice [root]
  InvoiceLine [1..*]
    ItemInformation [1..1]
      ItemAttributes [0..*]
```

The required **occurrence dimensions** are d_Invoice for Invoice; d_Invoice and d_InvoiceLine for InvoiceLine; d_Invoice and d_InvoiceLine for ItemInformation; and d_Invoice, d_InvoiceLine, and d_ItemAttributes for ItemAttributes. The Framework MUST NOT generate d_ItemInformation for this hierarchy.

Because hypercubes are closed, adding d_ItemInformation to the ItemAttributes cube would require a dimension that **Structured CSV** does not emit and would make an otherwise valid fact dimensionally invalid. This is a model-consistency requirement, not a processor-specific workaround.

7 OIM taxonomy artefacts

7.1 OIM module schema

For each module M used by the **formal HMD** set, the generator MUST write M/M-oim-<V>.xsd. The schema MUST contain the module-owned Attribute item declarations and p_ Class primary-item anchors required by the OIM binding. It MUST contain no Tuple C/R declarations and no Tuple structural complexTypes, and it MUST be independently XML Schema-valid.

7.2 OIM labels

The generator MUST write OIM-specific label linkbases at <module>/label/<module>-oim-lab-en-<V>.xml and <module>/label/<module>-oim-lab-<language>-<V>.xml. Their locators MUST resolve to concepts in <module>-oim-<V>.xsd. Attribute labels preserve the semantic labels of the corresponding Attribute QName; p_ Class anchors receive labels suitable for navigation.

7.3 OIM module presentation linkbase

For each module M, the generator MUST write M/M-oim-pre-<V>.xml. Each Class owned by M is an OIM presentation-tree starting candidate and is represented by p_M_<local_name>. Attribute descendants are represented by their OIM item concepts. Reference rows are traversed transparently and do not appear as OIM presentation concepts.

Traversal MAY cross module boundaries. Cross-module locators MUST resolve to the appropriate concept in the owning <module>-oim-<V>.xsd. Duplicate locators and duplicate presentationArc relationships within one presentationLink MUST NOT be serialized.

7.4 OIM Presentation Linkbase Set

The **OIM Presentation Linkbase Set** comprises the module OIM presentation linkbases published for the package. For an **HMD root**ed in module P, the **OIM message entry point** discovers P/P-oim-pre-<V>.xml. That root-module presentation contains the complete OIM-visible presentation tree reachable from the selected **HMD root**, with R rows traversed transparently.

The **OIM Presentation Linkbase Set** is distinct from the Tuple Presentation Linkbase Set because Class presentation concepts differ: Tuple uses C/R Tuple concepts, while OIM uses p_ Class anchors and no R presentation concept.

7.5 OIM message entry point and DTS separation

The **OIM message entry point** MUST be named <P>-all-oim-<V>.xsd and placed in oim/<HMD>. It MUST import every required <module>-oim-<V>.xsd, discover the OIM-specific module labels, the root-module OIM presentation linkbase, and the HMD-specific dimensional definition linkbase.

An OIM DTS MUST NOT discover Tuple <module>-<V>.xsd schemas or tuple/<HMD>/<module>-content-<V>.xsd schemas. This separation is required because Tuple C/R declarations depend on structural complexTypes that have no role in the OIM binding.

7.6 Dimensional definition linkbase

The HMD dimensional linkbase MUST be named <P>-all-dim-<V>.xml and placed beside the OIM message entry point. Its locators for p_ and Attribute concepts MUST resolve to module OIM schemas; locators for h_ concepts and applicable occurrence-key d_ concepts MUST resolve to the HMD OIM entry-point schema. It MUST contain the closed cube, occurrence-key-only hypercube-dimension, domain-member, and targetRole relationships defined in Clause 6.

7.7 Linkroles and DTS discovery

Each custom dimensional role URI MUST be declared by one discoverable link:roleType and referenced by link:roleRef from the definition linkbase. xlink:role identifies the dimensional network; the dimension arcroles identify all, **hypercube**-dimension, and domain-member relationships; xbrldt:**targetRole** continues traversal into a child-Class role.

8 xBRL-CSV metadata and data tables

8.1 Separation from taxonomy generation

The formal taxonomy package establishes OIM concepts, presentation, and dimensional networks. xBRL-CSV JSON metadata and **Structured CSV** data are report artefacts generated for a selected HMD and MUST NOT alter the taxonomy package or **formal HMD**.

8.2 documentInfo

The JSON metadata MUST declare documentType https://xbrl.org/2021/xbrl-csv, the selected HMD OIM taxonomy entry point, and every namespace used by concepts, dimensions, units, entities, and languages. The JSON metadata file is the xBRL-CSV entry document and the referenced **Structured CSV** file uses the same base report name.

```
"taxonomy": ["../../taxonomy/oim/cor_accountingEntries/cor-all-oim-<V>.xsd"]
...
"tables": {
  "structured": {
    "template": "structured",
    "url": "accountingEntries-journalEntry.csv"
  }
}
```

8.3 Occurrence-key columns

The table template MUST include occurrence-key columns only for the root Class and descendant Classes whose effective maximum occurrence is greater than one. Each occurrence-key column binds one-to-one to the corresponding HMD-specific typed dimension d_Q, and an OIM **occurrence dimension** MUST NOT exist for a Class without a corresponding **Structured CSV** occurrence-key column. Parent occurrence keys are retained on child rows where needed to preserve containment.

8.4 Property columns

Each reported Attribute column MUST set the concept dimension to the Attribute QName in the applicable OIM module schema. The concept's XBRL item type is resolved by explicit datatype binding and does not replace the HMD semantic datatype. Where a **Value Domain** binding applies, it constrains the fact value independently of **occurrence dimensions**. Monetary columns MUST set an appropriate unit, normally an ISO 4217 QName. A pure fact need not supply the Unit Core Dimension; the example metadata therefore does not encode xbrli:pure as an explicit unit for a pure value.

8.5 One Structured CSV per report

The examples use one rectangular **Structured CSV** for each report. Different Class occurrence types may be represented as informative rows in the same CSV. A singular Class may therefore have an informative row even though it has no occurrence-key column of its own; that row retains every applicable occurrence-key ancestor key. The record_type column is informative and is not a normative semantic identity mechanism. Multiple normalized CSV tables are not required to preserve the HMD hierarchy because containment is represented by the occurrence-key typed dimensions.

8.6 Reading a Structured CSV row through the cube

Each populated property value becomes an Attribute fact. Its concept aspect comes from the property-column metadata and its dimensional aspects come from the occurrence-key columns required by the applicable closed cube. A nested repeated Child row retains the occurrence keys of its occurrence-key ancestor lineage and its own Child occurrence key, thereby preserving parent-scoped occurrence identity. A singular intervening Class remains in the semantic hierarchy but contributes no occurrence-key column or typed dimension.

8.7 Report validation

An xBRL-CSV report MUST validate against the OIM specification, the selected HMD taxonomy entry point, datatype and unit rules, closed-cube constraints, typed-dimension requirements, and any additional business validation artefacts applicable to the report.

8.8 Shared `gen` module, Value Domains, and Structured CSV

The OIM taxonomy assembly MUST use the same shared generated `gen` taxonomy module as the Tuple assembly. The `gen` module contains binding-neutral reusable semantic definitions. OIM primary items, hypercubes, dimensions, dimensional relationships, and **Structured CSV** occurrence structures are binding-specific and MUST NOT be introduced into the semantic `gen` module.

The HMD `datatype` remains a semantic representation term. The XBRL item type is selected separately by an approved explicit occurrence or concept override followed by the approved default semantic-datatype mapping. An unresolved or `review-required` datatype is a generation error; no string fallback or name-based inference is permitted.

A **Value Domain** constrains the permitted semantic value of a fact and is separate from datatype and occurrence identity. The Part 1 occurrence-key classification derived from root status and multiplicity determines **Structured CSV** occurrence grouping and OIM typed dimensions. A **Value Domain** or EE 1.0 realisation MUST NOT create, merge, or identify an occurrence group, and an occurrence Class MUST NOT be modelled through an EE 1.0 **Value Domain**.

Where an **enumeration Value Domain** applies, one populated **Structured CSV** property cell represents one reported fact and one scalar **Value Domain** member. Extensible Enumerations 1.0 is the intended realisation where the approved binding requires an extensible enumeration; Extensible Enumerations 2.0 multiple-value semantics are not required for Framework **Value Domain** facts.

Legacy, standard, and user-defined **Value Domains** retain separate identities. Legacy values MUST NOT be silently translated to standard values during reporting or validation. Authority, vocabulary identifier, edition or version, subset, provenance, and rights status belong to the **Value Domain** and taxonomy binding, not to every **Structured CSV** cell.

A **Value Domain** crosswalk preserves both source and target **Value Domain** identities; use of a crosswalk is an explicit transformation operation. When the HMD `value_domain` requires an extensible enumeration, Extensible Enumerations 1.0 domain/member resources and applicable concept bindings MUST be generated deterministically from the authoritative **Value Domain**, Value, overlay, and datatype inputs. Extensible Enumerations 2.0 multiple-value semantics are not required.

For an EE1-bound fact concept, the generated declaration uses the Extensible Enumerations 1.0 enumeration item type, the generated **Value Domain** head and deterministic linkrole, with the domain head not directly usable (`enum:headUsable="false"`). The reported fact value is the QName of one permitted member, and the same governed **Value Domain** concepts are shared by the Tuple and OIM taxonomy assemblies.

The formal HMD remains an 18-column contract and contains no `domain_name` column. Column 9 is `value_domain`, preserved unchanged from the approved LHM reviewed. The generated candidate LHM retains `association_role` in column 9; the controlled LHM-review step replaces that field with the reviewed `value_domain` assignment. The **Value Domain** is binding metadata for the fact concept and does not create an occurrence identity or alter the HMD `datatype`.

8.9 Implemented occurrence and Value Domain rules

The current maintained Generator implements the occurrence-key rule required by this Part: the HMD root is an **occurrence-key Class**; a non-root Class is an **occurrence-key Class** only when its effective maximum occurrence is greater than one. Non-root `0..1`, `1`, and `1..1` Classes do not create OIM **occurrence dimensions**.

The **Structured CSV** and OIM taxonomy processing MUST use the same occurrence-key classification derived from root status and effective multiplicity. Selector-qualified physical variants MAY cause a selector-effective multiplicity to be derived for taxonomy generation, but this MUST NOT rewrite the accepted semantic HMD multiplicity. The derived source/effective multiplicity and selector evidence MUST remain auditable.

The **Value Domain** pipeline is authoritative **Value Domain** definition → reviewed LHM `value_domain` assignment → HMD-for-taxonomy preservation → EE1 domain/member generation → applicable concept binding. Missing domains, unresolved values, invalid domain assignment, or a **Value Domain** applied to a non-Attribute row are generation errors; generation MUST NOT silently fall back to an ungoverned code type.

9 Cross-binding equivalence

Tuple and OIM bindings are equivalent when every effective HMD Attribute occurrence maps to the same semantic_path and expanded concept QName, and when Class and Association scopes preserve the same ancestor sequence and multiplicities. Tuple nesting preserves every effective Class. OIM **occurrence dimensions** preserve only **occurrence-key Classes**, while p_Q, h_Q, roles, domain-member hierarchy, and targetRole preserve the broader semantic Class hierarchy. These physical mechanisms MUST NOT be compared by XML shape or by requiring one dimension for every Tuple Class.

HMD construct	Part 2 Tuple binding	Part 3 OIM binding
Class (C)	Tuple element and complex type	Primary item, closed hypercube, and extended role for every Class; typed **occurrence dimension** only for an occurrence-key Class
Reference (R)	Structural Tuple element	Transparent traversal unless an explicit reference-fact rule applies
Attribute (A)	XBRL item child of structural tuple	Fact concept in the applicable Class cube
Hierarchy	xs:sequence and occurrence constraints	definition relationships, dimensions, and **targetRole**
Identity	Expanded QName plus physical occurrence path	Expanded QName plus occurrence-dimension values

10 accountingEntries baseline and businessTransactions additional HMD (informative)

The accountingEntries **HMD root**ed at gl-cor:accountingEntries generates oim/cor_accountingEntries/cor-all-oim-<V>.xsd and cor-all-dim-<V>.xml. Its entry point imports the required module OIM schemas and discovers cor/cor-oim-pre-<V>.xml.

The businessTransactions **HMD rooted** at gl-btx:businessTransactions independently generates oim/btx_businessTransactions/btx-all-oim-<V>.xsd and btx-all-dim-<V>.xml. Its entry point imports the required module OIM schemas and discovers btx/btx-oim-pre-<V>.xml.

Adding the businessTransactions **HMD** does not change the accountingEntries root, dimensional network, or entry point. Reused Attribute QNames preserve their semantic identity while the two root-specific **HMD** DTSs retain independent dimensional constraints.

Validation scope is one **HMD for taxonomy** / one OIM DTS. Each root-specific HMD has its own dimensional network and closure. A QName reused in another root-specific DTS does not create a cross-DTS equality requirement unless it is deliberately governed as the same shared declaration.

11 Formal package assembly

11.1 Formal package tree

```
taxonomy/
  gen/
    gl-gen-<V>.xsd
  <M>/
    <M>-<V>.xsd
    <M>-pre-<V>.xml
    <M>-oim-<V>.xsd
    <M>-oim-pre-<V>.xml
    label/
      <M>-lab-en-<V>.xml
      <M>-lab-<language>-<V>.xml
      <M>-oim-lab-en-<V>.xml
      <M>-oim-lab-<language>-<V>.xml
  tuple/
    <HMD>/
      <P>-all-<V>.xsd
```

```
        <M>-content-<V>.xsd
    oim/
      <HMD>/
        <P>-all-oim-<V>.xsd
        <P>-all-dim-<V>.xml
  NOTE In this directory tree, terms enclosed in angle brackets are metavariables and
  are replaced as follows:
     • <M> is the module identifier of a taxonomy module, for example cor, btx,
       bus, lnk, or taf.
     • <HMD> is the deterministic HMD identifier formed from the root module
       identifier, an underscore, and the root local_name, for example
       cor_accountingEntries or btx_businessTransactions.
     • <P> is the root-module identifier used as the filename prefix for the HMD
       entry point and its HMD-specific linkbases, for example cor for
       cor_accountingEntries and btx for btx_businessTransactions.
     • <V> is the taxonomy version expressed as a date in YYYY-MM-DD form, for
       example 2026-12-31.
     • <language> is the language identifier used for an additional-language label
       linkbase, for example ja.
  The angle brackets are not part of the actual directory or file names.
```

The OIM files governed by this Part are the <module>-oim-<V>.xsd schemas, <module>-oim-pre-<V>.xml presentation linkbases, OIM-specific labels, and the root-specific OIM entry point and dimensional linkbase. Tuple-specific components are governed by Part 2.

11.2 Version and namespaces

The package version V MUST be an explicit date in YYYY-MM-DD form. The module namespace for M MUST be https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/<M>. The palette namespace for HMD-specific OIM h_/d_ resources MUST be https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/plt.

11.3 Deterministic HMD identifier

For a root XPath terminal gl-M:L, the HMD identifier MUST be M_L and unique within the **HMD for taxonomy** input set. The current examples use cor_accountingEntries and btx_businessTransactions.

11.4 All-or-nothing publication

A **formal package** is conformant only when every **formal HMD** is valid and all required Tuple and OIM components form valid complete DTSs. **OIM module schema**s MUST be independently valid, OIM entry points MUST resolve every local file and fragment reference, and the OIM DTS MUST remain free of Tuple module/content schemas.

12 Validation and conformance

12.1 Taxonomy conformance

- Every **OIM module schema** and root-specific OIM entry point MUST be valid and discoverable.
- Every OIM presentation locator MUST resolve to an xbrli:item concept in an **OIM module schema**; Class locators use p_ primary items and R rows are transparent.
- Every Class role MUST contain the required **primary item**, closed cube, ancestor dimensions, Attribute memberships, and **targetRole** transitions.
- OIM entry points and dimensional linkbases MUST NOT discover Tuple module or content schemas.
- Duplicate presentation locators and presentationArcs MUST be absent.

- The accountingEntries and businessTransactions HMDs MUST remain independent root-specific OIM DTSs.
- The package generation result MUST be deterministic.
- Every OIM **occurrence dimension** MUST correspond to one applicable **Structured CSV** occurrence-key column, and every applicable **Structured CSV** occurrence-key column MUST correspond to its OIM **occurrence dimension**.
- Every closed hypercube MUST contain only the occurrence-key dimensions on the applicable ancestor path, plus the current Class dimension when the current Class is occurrence-key.

12.2 Minimum report tests

- one and multiple root/Class occurrences;
- nested and repeating Class occurrences;
- Attributes reached through a Reference row;
- **targetRole** navigation with required ancestor **occurrence dimensions**;
- monetary, pure, and non-monetary Attribute columns;
- rejection of an undeclared dimension by a closed cube;
- separate accountingEntries and businessTransactions JSON/Structured-CSV pairs;
- validation that each JSON metadata document resolves only the appropriate OIM DTS; and
- a mandatory repeat-to-singular-to-repeat regression in which InvoiceLine [1..*] contains ItemInformation [1..1], which contains ItemAttributes [0..*]: the report and taxonomy MUST include d_InvoiceLine and d_ItemAttributes, MUST NOT include d_ItemInformation, and MUST preserve each nested ItemAttributes occurrence within its parent InvoiceLine scope.

13 Normative output-name summary

Component	Pattern
OIM module schema	<module>/<module>-oim-<V>.xsd
OIM module presentation	<module>/<module>-oim-pre-<V>.xml
OIM base-language labels	<module>/label/<module>-oim-lab-en-<V>.xml
OIM local-language labels	<module>/label/<module>-oim-lab-<language>-<V>.xml
OIM message entry point	oim/<HMD>/<P>-all-oim-<V>.xsd
Dimensional linkbase	oim/<HMD>/<P>-all-dim-<V>.xml
xBRL-CSV metadata	report-defined JSON referencing one selected OIM message entry point
Structured CSV	one report data file referenced by the JSON metadata in the examples

14 xBRL-CSV instance examples (informative)

14.1 Example files

The current example package provides one JSON metadata document and one **Structured CSV** file for each root-specific report. The JSON and CSV use the same base name.

Example	JSON metadata	Structured CSV
Accounting Entries journal entry	ids/oim/ accountingEntries-journalEntry.json	ids/oim/ accountingEntries-journalEntry.csv
Business Transactions vendor invoice	ids/oim/ businessTransactions-vendorInvoice.json	ids/oim/ businessTransactions-vendorInvoice.csv

14.2 Accounting Entries example

accountingEntries-journalEntry.json references taxonomy/oim/cor_accountingEntries/cor-all-oim-<V>.xsd. Its table template binds occurrence-key columns such as dAccountingEntries, dEntryHeader, and dEntryDetail to the corresponding plt:d_* typed dimensions. Property columns bind to cor Attribute QNames.

```
{
  "documentInfo": {
    "documentType": "https://xbrl.org/2021/xbrl-csv",
    "namespaces": {
      "plt": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/plt",
      "cor": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/cor",
      "bus": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/bus",
      "btx": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/btx",
      "lnk": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/lnk",
      "taf": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/taf",
      "xbrli": "http://www.xbrl.org/2003/instance",
      "iso4217": "http://www.xbrl.org/2003/iso4217",
      "ent": "https://example.com/entity/"
    },
    "taxonomy": [
      "../../taxonomy/oim/cor_accountingEntries/cor-all-oim-2026-12-31.xsd"
    ]
  },
  "dimensions": {
    "entity": "ent:EXAMPLE-BUYER-001",
    "period": "2026-08-11T00:00:00"
  },
  "tableTemplates": {
    "structured": {
      "columns": {
        "record_type": {
          "comment": true
        },
        "dAccountingEntries": {},
        "dEntryHeader": {},
        "dEntryDetail": {},
        "documentType": {
          "dimensions": {
            "concept": "cor:documentType"
          }
        },
        "creationDate": {
          "dimensions": {
            "concept": "cor:creationDate"
          }
        },
        "creator": {
          "dimensions": {
            "concept": "cor:creator"
```

```
      }
    },
    "documentComment": {
      "dimensions": {
        "concept": "cor:documentComment"
      }
    },
    "defaultCurrency": {
      "dimensions": {
        "concept": "cor:defaultCurrency"
      }
    },
    "entityPartyName": {
      "dimensions": {
        "concept": "cor:entityPartyName"
      }
    },
    "entityPartyId": {
      "dimensions": {
        "concept": "cor:entityPartyId"
      }
    },
    "datePosted": {
      "dimensions": {
        "concept": "cor:datePosted"
      }
    },
    "headerEntryDate": {
      "dimensions": {
        "concept": "cor:headerEntryDate"
      }
    },
    "sourceJournal": {
      "dimensions": {
        "concept": "cor:sourceJournal"
      }
    },
    "entryId": {
      "dimensions": {
        "concept": "cor:entryId"
      }
    },
    "entryDescription": {
      "dimensions": {
        "concept": "cor:entryDescription"
      }
    },
    "totalDebits": {
      "dimensions": {
        "concept": "cor:totalDebits",
        "unit": "iso4217:JPY"
      },
      "decimals": 0
    },
    "totalCredits": {
      "dimensions": {
        "concept": "cor:totalCredits",
        "unit": "iso4217:JPY"
      },
```

```
            "decimals": 0
          },
          "lineNumber": {
            "dimensions": {
              "concept": "cor:lineNumber"
            }
          },
          "detailDocumentType": {
            "dimensions": {
              "concept": "cor:detailDocumentType"
            }
          },
          "detailDocumentNumber": {
            "dimensions": {
              "concept": "cor:detailDocumentNumber"
            }
          },
          "documentDate": {
            "dimensions": {
              "concept": "cor:documentDate"
            }
          },
          "detailDescription": {
            "dimensions": {
              "concept": "cor:detailDescription"
            }
          },
          "monetaryAmount": {
            "dimensions": {
              "concept": "cor:monetaryAmount",
              "unit": "iso4217:JPY"
            },
            "decimals": 0
          },
          "currency": {
            "dimensions": {
              "concept": "cor:currency"
            }
          },
          "debitCreditIndicator": {
            "dimensions": {
              "concept": "cor:debitCreditIndicator"
            }
          }
        },
        "dimensions": {
          "plt:d_cor_accountingEntries": "$dAccountingEntries",
          "plt:d_cor_entryHeader": "$dEntryHeader",
          "plt:d_cor_entryDetail": "$dEntryDetail"
        }
      }
    },
    "tables": {
      "structured": {
        "template": "structured",
        "url": "accountingEntries-journalEntry.csv"
      }
    }
  }
```

The **Structured CSV** contains DocumentInformation, EntityParty, EntryHeader, and repeated EntryDetail rows. The three EntryDetail rows record JPY 10,000 expense, JPY 1,000 input tax, and JPY 11,000 accounts payable, linked by the occurrence keys AE-001 and AE-HDR-001.

```
record_type,dAccountingEntries,dEntryHeader,dEntryDetail,documentType,creationDate,
creator,documentComment,defaultCurrency,entityPartyName,entityPartyId,datePosted,he
aderEntryDate,sourceJournal,entryId,entryDescription,totalDebits,totalCredits,lineN
umber,detailDocumentType,detailDocumentNumber,documentDate,detailDescription,moneta
ryAmount,currency,debitCreditIndicator
DocumentInformation,AE-001,,,journalEntry,2026-08-11T00:00:00,XBRL GL Next
sample,Vendor invoice recognition sample,iso4217:JPY,,,,,,,,,,,,,,,,
EntityParty,AE-001,,,,,,,,Example Buyer Co.,EXAMPLE-BUYER-001,,,,,,,,,,,,,,
EntryHeader,AE-001,AE-HDR-001,,,,,,,,2026-08-11T00:00:00,2026-08-
11T00:00:00,AP,JE-2026-0811-001,Record vendor invoice INV-2026-0811-
001,11000,11000,,,,,,,
EntryDetail,AE-001,AE-HDR-001,AE-DET-1,,,,,,,,,,,,,,,1,invoice,INV-2026-0811-
001,2026-08-11T00:00:00,Office supplies expense,10000,iso4217:JPY,D
EntryDetail,AE-001,AE-HDR-001,AE-DET-2,,,,,,,,,,,,,,,2,invoice,INV-2026-0811-
001,2026-08-11T00:00:00,Input consumption tax,1000,iso4217:JPY,D
EntryDetail,AE-001,AE-HDR-001,AE-DET-3,,,,,,,,,,,,,,,3,invoice,INV-2026-0811-
001,2026-08-11T00:00:00,Accounts payable,11000,iso4217:JPY,C
```

14.3 Business Transactions example

businessTransactions-vendorInvoice.json references taxonomy/oim/btx_businessTransactions/btx-all-oim-<V>.xsd. The **Structured CSV** uses one table containing TransactionDocumentInformation, EntityParty, TransactionHeader, HeaderTax, HeaderAmountStructure, TransactionDetail, and DetailAmountStructure rows. Parent keys are repeated on child rows to preserve containment.

```
{
  "documentInfo": {
    "documentType": "https://xbrl.org/2021/xbrl-csv",
    "namespaces": {
      "plt": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/plt",
      "cor": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/cor",
      "bus": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/bus",
      "btx": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/btx",
      "lnk": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/lnk",
      "taf": "https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/taf",
      "xbrli": "http://www.xbrl.org/2003/instance",
      "iso4217": "http://www.xbrl.org/2003/iso4217",
      "ent": "https://example.com/entity/"
    },
    "taxonomy": [
      "../../taxonomy/oim/btx_businessTransactions/btx-all-oim-2026-12-31.xsd"
    ]
  },
  "dimensions": {
    "entity": "ent:EXAMPLE-BUYER-001",
    "period": "2026-08-11T00:00:00"
  },
  "tableTemplates": {
    "structured": {
      "columns": {
        "record_type": {
          "comment": true
        },
        "dBusinessTransactions": {},
        "dTransactionHeader": {},
```

```
            "dHeaderTax": {},
            "dHeaderAmountStructure": {},
            "dTransactionDetail": {},
            "dDetailAmountStructure": {},
            "transactionDocumentType": {
              "dimensions": {
                "concept": "cor:transactionDocumentType"
              }
            },
            "transactionCreationDate": {
              "dimensions": {
                "concept": "cor:transactionCreationDate"
              }
            },
            "transactionCreator": {
              "dimensions": {
                "concept": "cor:transactionCreator"
              }
            },
            "transactionDocumentComment": {
              "dimensions": {
                "concept": "cor:transactionDocumentComment"
              }
            },
            "transactionDefaultCurrency": {
              "dimensions": {
                "concept": "cor:transactionDefaultCurrency"
              }
            },
            "entityPartyName": {
              "dimensions": {
                "concept": "cor:entityPartyName"
              }
            },
            "entityPartyId": {
              "dimensions": {
                "concept": "cor:entityPartyId"
              }
            },
            "datePosted": {
              "dimensions": {
                "concept": "btx:datePosted"
              }
            },
            "headerDocumentType": {
              "dimensions": {
                "concept": "btx:headerDocumentType"
              }
            },
            "headerInvoiceType": {
              "dimensions": {
                "concept": "btx:headerInvoiceType"
              }
            },
            "headerDocumentNumber": {
              "dimensions": {
                "concept": "btx:headerDocumentNumber"
              }
            },
```

```
            "headerMaturityDateOrDateDue": {
              "dimensions": {
                "concept": "btx:headerMaturityDateOrDateDue"
              }
            },
            "headerPaymentTerms": {
              "dimensions": {
                "concept": "btx:headerPaymentTerms"
              }
            },
            "taxAuthority": {
              "dimensions": {
                "concept": "btx:taxAuthority"
              }
            },
            "amountOfTaxes": {
              "dimensions": {
                "concept": "btx:amountOfTaxes",
                "unit": "iso4217:JPY"
              },
              "decimals": 0
            },
            "basisForTaxation": {
              "dimensions": {
                "concept": "btx:basisForTaxation",
                "unit": "iso4217:JPY"
              },
              "decimals": 0
            },
            "taxPercentageRate": {
              "dimensions": {
                "concept": "btx:taxPercentageRate"
              },
              "decimals": 2
            },
            "taxCategory": {
              "dimensions": {
                "concept": "btx:taxCategory"
              }
            },
            "headerAmount": {
              "dimensions": {
                "concept": "btx:headerAmount",
                "unit": "iso4217:JPY"
              },
              "decimals": 0
            },
            "headerAurrencyCode": {
              "dimensions": {
                "concept": "btx:headerAurrencyCode"
              }
            },
            "headerAmountType": {
              "dimensions": {
                "concept": "btx:headerAmountType"
              }
            },
            "lineNumber": {
              "dimensions": {
```

```
            "concept": "btx:lineNumber"
          }
        },
        "documentType": {
          "dimensions": {
            "concept": "btx:documentType"
          }
        },
        "documentNumber": {
          "dimensions": {
            "concept": "cor:documentNumber"
          }
        },
        "documentDate": {
          "dimensions": {
            "concept": "btx:documentDate"
          }
        },
        "detailDescription": {
          "dimensions": {
            "concept": "btx:detailDescription"
          }
        },
        "amount": {
          "dimensions": {
            "concept": "btx:amount",
            "unit": "iso4217:JPY"
          },
          "decimals": 0
        },
        "currencyCode": {
          "dimensions": {
            "concept": "btx:currencyCode"
          }
        },
        "detailAmountType": {
          "dimensions": {
            "concept": "btx:detailAmountType"
          }
        }
      },
      "dimensions": {
        "plt:d_btx_businessTransactions": "$dBusinessTransactions",
        "plt:d_btx_transactionHeader": "$dTransactionHeader",
        "plt:d_btx_headerTax": "$dHeaderTax",
        "plt:d_btx_headerAmountStructure": "$dHeaderAmountStructure",
        "plt:d_btx_transactionDetail": "$dTransactionDetail",
        "plt:d_btx_detailAmountStructure": "$dDetailAmountStructure"
      }
    }
  },
  "tables": {
    "structured": {
      "template": "structured",
      "url": "businessTransactions-vendorInvoice.csv"
    }
  }
}
```

The example represents vendor invoice INV-2026-0811-001, with JPY 10,000 taxable basis, JPY 1,000 tax, and JPY 11,000 header amount. The same invoice number also appears in the informative Tuple Accounting Entries example, demonstrating cross-root business traceability without combining the two HMDs into one DTS.

```
record_type,dBusinessTransactions,dTransactionHeader,dHeaderTax,dHeaderAmountStruct
ure,dTransactionDetail,dDetailAmountStructure,transactionDocumentType,transactionCr
eationDate,transactionCreator,transactionDocumentComment,transactionDefaultCurrency
,entityPartyName,entityPartyId,datePosted,headerDocumentType,headerInvoiceType,head
erDocumentNumber,headerMaturityDateOrDateDue,headerPaymentTerms,taxAuthority,amount
OfTaxes,basisForTaxation,taxPercentageRate,taxCategory,headerAmount,headerAurrencyC
ode,headerAmountType,lineNumber,documentType,documentNumber,documentDate,detailDesc
ription,amount,currencyCode,detailAmountType
TransactionDocumentInformation,BT-001,,,,,,vendorInvoice,2026-08-11T00:00:00,XBRL
GL Next sample,Vendor invoice sample corresponding to JE-2026-0811-
001,iso4217:JPY,,,,,,,,,,,,,,,,,,,,,,
EntityParty,BT-001,,,,,,,,,,Example Buyer Co.,EXAMPLE-BUYER-
001,,,,,,,,,,,,,,,,,,,,
TransactionHeader,BT-001,BT-HDR-001,,,,,,,,,,,2026-08-
11T00:00:00,invoice,vendorInvoice,INV-2026-0811-001,2026-09-10T00:00:00,Net 30
days,,,,,,,,,,,,,,,
HeaderTax,BT-001,BT-HDR-001,BT-TAX-
001,,,,,,,,,,,,,,,,JP,1000,10000,0.10,standard,,,,,,,,,,
HeaderAmountStructure,BT-001,BT-HDR-001,,BT-HAMT-
001,,,,,,,,,,,,,,,,,,,,11000,JPY,invoiceTotal,,,,,,,,
TransactionDetail,BT-001,BT-HDR-001,,,BT-DET-
001,,,,,,,,,,,,,,,,,,,,,,1,invoice,INV-2026-0811-001,2026-08-11T00:00:00,Office
supplies,,,
DetailAmountStructure,BT-001,BT-HDR-001,,,BT-DET-001,BT-DAMT-
001,,,,,,,,,,,,,,,,,,,,,,,,,,,,10000,JPY,lineNetAmount
```

14.4 Concepts, dimensions, and presentation

Reported facts use Attribute item concepts from the module OIM schemas. Class hierarchy is represented by p_ primary-item anchors, roles, domain-member/targetRole relationships, and closed hypercubes. Typed **occurrence dimensions** represent only **occurrence-key Classes**. The module-level p_ primary items are not the Attribute facts populated by the example CSV. Reference rows are not represented as facts and are traversed transparently.

14.5 Unit handling

Monetary columns specify an ISO 4217 unit such as iso4217:JPY. Where an example value is pure, the Unit Core Dimension may be omitted rather than encoding xbrli:pure as an explicit unit. This keeps the metadata aligned with xBRL-CSV aspect rules.

Appendix A
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)
SCHEMA-1	World Wide Web Consortium (W3C). "XML Schema Part 1: Structures" (See http://www.w3.org/TR/xmlschema-1/)
XPath	World Wide Web Consortium (W3C), XML Path Language (XPath) 2.0 (Second Edition), W3C Recommendation, 14 December 2010. (See https://www.w3.org/TR/xpath20/)
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	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/)
XLINK	World Wide Web Consortium (W3C). "XML Linking Language (XLink) Version 1.0." Steve DeRose, Eve Maler, and David Orchard. (See http://www.w3.org/TR/xlink/)
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)

Appendix B
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 C
Document history

2026-09-15 accepted alignment: This section records the current accepted project specification.

Topic	Accepted specification and boundary
Project module namespace	The namespace of module M is exactly https://www.xbrl.or.jp/taxonomy/xbrl-gl-next/M. The URI contains neither an experimental segment nor a date. Namespace prefixes, versioned physical filenames, and directory paths are separate identifiers and do not alter the namespace URI. External XML, XBRL, XLink, ISO, and historical 2015 XBRL GL namespaces remain unchanged.
Model authority and pipeline	The authoritative authored source is the FSM. Specialisation deterministically produces the BSM. Graph Walk produces the candidate LHM. Human review controls the reviewed LHM. Post-Graph Walk produces the root-specific HMDs. The HMD semantic_path, hierarchy, datatype, multiplicity, and value_domain govern Tuple and OIM generation.
Root-specific DTSs	Accounting Entries and Business Transactions are expanded from different root HMDs. Each root is used and validated through its matching Tuple or OIM entry point. The split layout is mandatory; same-named files are not copied over one another and the two DTSs are not mixed without a separate integration design.
Accepted structural differences	Fourteen tracked differences among same-module and same-name declarations are accepted as ROOT_HMD_SPECIFIC_STRUCTURAL_DIFFERENCE. Exact equality of root-derived type, structure, or multiplicity is not required across AE and BT. Each declaration remains subject to its own accepted root HMD and DTS.
OIM binding	A selected root HMD is realised using Class primary items, closed cubes, typed occurrence dimensions for occurrence-key Classes, domain-member hierarchy, and targetRole transitions. targetRole is the XBRL Dimensions navigation attribute; targetRef is not substituted for it.
xBRL-CSV boundary	Formal xBRL-CSV uses JSON metadata as its entry point. The metadata identifies concept QNames, entity, period, unit, dimensions, the same-basename CSV table, and the matching root-specific OIM DTS. A long-form Structured CSV is an intermediate UADC representation and is not itself substituted for the formal JSON entry point.
Occurrence and split validation	Occurrence values preserve repeated Class identity within the selected HMD. AE and BT OIM dimensional networks are validated separately; their accepted 14 structural differences do not require commonisation or multiplicity relaxation.

12 August 2026 v7: Defined the initial-edition OIM module Palette with standalone module OIM schemas, module-level p_ Class **primary items**, binding-specific **OIM Presentation Linkbase Set**, explicit Tuple/OIM DTS separation, final package naming, and informative single-Structured-CSV instance examples for Accounting Entries and Business Transactions.

20 August 2026 v8: Added the shared `gen` taxonomy module and **Value Domain** architecture; clarified semantic datatype and XBRL item type separation; preserved distinct **Value Domain** identities; and defined scalar **Structured CSV Value Domain** treatment and the future implementation boundary.

21 August 2026 v9: Restricted d_Q generation and closed-cube ancestry to **occurrence-key Classes** derived from root status and multiplicity; established one-to-one **Structured CSV**/OIM occurrence-key alignment; preserved p_Q, h_Q, roles, hierarchy, and targetRole for singular Classes; separated **Value Domain**/EE 1.0 from occurrence identity; and recorded the pending Generator changes.

Date	Details
8 August 2026	Rewritten around the formal 18-column HMD contract and deterministic OIM generation; specified closed cubes, **targetRole**, xBRL-CSV metadata boundaries, HMD-directed package assembly, accountingEntries baseline, and businessTransactions additional-HMD example.
4 August 2026	Previous collaborative review draft.
8 August 2026	v2: Added shared DTS and linkbase discovery, module-concept locator reuse, dimensional roleType/roleRef/linkrole rules, and the relationship between OIM definition networks and the shared HMD presentation view.
8 August 2026	v3: Added the common terminology statement and clarified how the OIM entry point reuses the module palette illustrated in Part 2 while retaining an independent dimensional definition network.
8 August 2026	v4: Added an intuitive explanation and Figure 1 showing how **primary items**, closed **hypercubes**, typed **occurrence dimensions**, primary-item members, and **targetRole**-connected Class roles represent the HMD hierarchy and bind **structured CSV** rows. Clarified that **targetRole**, not targetRef, is the XBRL Dimensions attribute and distinguished typed dimension values from explicit dimension members.
11 August 2026	v6: Aligned Part 3 with the accepted OIM package and report convention: common HMD-for-taxonomy input, <P>-all-oim/dim/pre filenames under oim/<HMD>, OIM-specific presentation extensions distinct from Tuple, no Tuple content-schema discovery, and one JSON metadata entry point plus one same-basename **Structured CSV** per example report. Clarified reusable cross-HMD Attribute declarations versus HMD-specific structural/Class occurrence resources.
28 August 2026	Phase 2C: aligned the HMD header with `value_domain`, recorded implemented EE1 **Value Domain** generation and occurrence-key rules, clarified selector-effective multiplicity without HMD rewriting, identified the sole self-contained Generator authority, and stated one-HMD/one-OIM-DTS validation scope.
29 August 2026	Renamed the former “LHM for taxonomy” terminology to “**HMD for taxonomy**”; aligned HMD column 9 with the reviewed-LHM `value_domain` authority; clarified deterministic EE1 implementation details; and limited cross-DTS equality to deliberately shared declarations while preserving root-specific OIM dimensional networks.
15 September 2026	Restored the accepted 2026-08-29 successor as the base; aligned project module namespaces with the stable URI; recorded root-HMD-specific AE/BT split conformance and the accepted 14 structural differences; aligned the UADC interface boundary and current semantic items where applicable.
6 November 2026	Revised the correspondence with the XBRL GL Next Requirements Specification to reflect the updated requirements.

Appendix D
Errata corrections in this document

This appendix records editorial corrections incorporated into this collaborative review draft. The entries have not been approved or endorsed by XBRL Japan, XBRL Europe, XBRL International, or any other organization.

No errata have been incorporated into this document.

Appendix E
(informative)
Correspondence with XBRL GL Next Requirements

Table E.1 maps the applicable requirements of the XBRL GL Next Requirements Specification to their implementation in this Part. The requirement identifiers and names are reproduced from that Specification. This correspondence is informative and does not create additional requirements or assert implementation conformance.

Table E.1 — Mapping of XBRL GL Next Requirements

Requirement ID	Requirement Name	Implementation in Part 3
GL-REQ-0001	Consistent business semantics across representations	Uses formal HMD-for-taxonomy inputs from Part 1 and preserves their semantic paths, reviewed datatypes, multiplicities and source traceability while generating a separate OIM binding (Clauses 4–5).
GL-REQ-0107	Business completeness and data quality	Requires validation of authoritative HMD rows, resolved QNames, dimensional and closed-cube constraints, taxonomy entry points and xBRL-CSV report metadata and facts (Clauses 4, 8.7 and 12).
GL-REQ-0108	Business transaction coverage and linkage	Generates independent accountingEntries and businessTransactions OIM entry points and dimensional networks, without combining their root-specific DTSs (Clauses 7.5 and 10).
GL-REQ-0201	Reusable business scope	Defines reusable OIM module schemas, primary-item anchors, labels and presentation networks, together with separate HMD-specific hypercubes, dimensions and message entry points (Clauses 5, 7 and 11).
GL-REQ-0201	Reusable business scope	Selects the root-specific OIM DTS and required module declarations from the formal HMD, keeping each root's own dimensional scope and taxonomy entry point (Clauses 7.5, 10 and 11).
GL-REQ-0301	Exchange-format independence	Provides the OIM-compatible binding of the common semantic model, with xBRL-CSV metadata and data tables distinguished from taxonomy generation (Clauses 2.1, 7–8).
GL-REQ-0301	Exchange-format independence	Defines Attribute fact concepts, reusable Class anchors, closed hypercubes, occurrence-key typed dimensions, presentation and targetRole networks and discoverable OIM DTSs (Clauses 5–8).
GL-REQ-0303	Identity and relationships of repeated business information	Defines cross-binding equivalence by semantic_path, expanded fact QName, ancestor scope and multiplicity, rather than by requiring OIM dimensional structure to resemble XML Tuple nesting (Clause 9).
GL-REQ-0303	Identity and relationships of repeated business information	Restricts typed occurrence dimensions to the root and repeating Classes, aligns them one-to-one with Structured CSV occurrence-key columns and preserves parent-scoped identity through closed cubes (Clauses 5–6 and 8.3).

Requirement ID	Requirement Name	Implementation in Part 3
GL-REQ-0401	Accessible examples and independent inspection	Includes Accounting Entries and Business Transactions xBRL-CSV JSON/CSV examples and specifies taxonomy and report-validation checks for their entry points and dimensional constraints (Clauses 12 and 14).
GL-REQ-0402	Demonstrable business interoperability	Specifies evidence of deterministic OIM generation, full DTS discovery, closed-cube consistency, dimension-column alignment and report validation through independent HMD roots (Clause 12).
GL-REQ-0501	Business code values and enumerations	Defines deterministic EE1 realisation of reviewed Value Domain assignments as one permitted member for each scalar fact, separate from occurrence dimensions (Clauses 8.8–8.9).
GL-REQ-0502	Reference to externally governed codes	Preserves governed legacy, standard and user-defined Value Domain identities and keeps external provenance, vocabulary versions, rights and crosswalk operations separate from report cells (Clause 8.8).