

Development

How the model of each UBL version is generated, corrected and tested, and how to add a new version

Anonymous · 13/10/2026

Development

Nothing under `src/V*/0asis` is written by hand: it is generated from the official OASIS XSD, corrected, and proven against the official examples. This page is for whoever maintains the package.

The Tools

Every tool in `tools/` takes an optional UBL version (`php tools/generate-oasis.php 2.2`) and does all of them without it. What they share, including where OASIS publishes what for each version, is in `tools/oasis-common.php`.

Tool	What it does
<code>generate-oasis.php</code>	Downloads the official XSD of the version into <code>resources/schemas/xsd/<version>/</code> (if it is not there already) and generates the model into <code>src/V<nn>/0asis/</code> with <code>xterr/php-ubl-generator</code> .
<code>patch-oasis.php</code>	Corrects the defects of that generated code (see below). Idempotent.
<code>fetch-oasis-examples.php</code>	Downloads the official XML examples of the version into <code>tests/fixtures/<version>/doc/xml/</code> .
<code>fetch-oasis-json.php</code>	Downloads the official JSON examples into <code>tests/fixtures/<version>/doc/json/</code> and the official JSON schemas into <code>resources/schemas/json/<version>/</code> (2.1, 2.2 and 2.3 only).
<code>generate-yaml-fixtures.php</code>	Writes the minimal YAML fixture of the documents of a version that do not have one. Never overwrites a file.
<code>patch-xterr-ubl.php</code>	Corrects defects of the runtime, <code>xterr/php-ubl</code> , in <code>vendor/</code> .

`composer run generate-oasis` runs the generator, the patch and the formatter for every version, and the result is deterministic.

Defects That Are Corrected

In the generated code (`patch-oasis.php`):

- The generator puts every class that is not `Cbc` into `Cac`, including the extension (`Ext`) and signature (`Sac`, `Sig`) ones: they are moved to the directory of their own XML namespace.
- A `cac:` element whose name is also a `cbc:` one (`Location`, `Condition`, `Status`, `Response`) is typed as the text of the `cbc:` one and loses its content. The places are found, not listed: there are two in 2.1 and eight from 2.4, and some are required.
- A list of a built-in type (`ElectronicPaymentUsageIndicator`, a list of booleans, from 2.2) is typed as a class that does not exist and guarded with an `instanceof` that never holds.
- Time elements without their format, the wildcard of the extension content (`xs:any`), the signature envelope, and the lists in the `validate*()` docblocks.
- Every class gets `ArraySerializableTrait` and its precomputed `ARRAY_METADATA`.

In the runtime (`patch-xterr-ubl.php`):

- Dates: the strict format falls back to the lenient parser, and the result is canonical.
- The unmapped children of an extension (a signature) are kept as XML that stands on its own, with the namespaces its ancestors declared and it uses, and written back as they were read, byte by byte (DOM reconciles what it imports, which turned the default namespace of a XAdES signature into a prefixed one).

Each patch checks the exact source it expects and fails loudly if it is not there.

Codes

The UBL XSD contain no enumerations: every code element is a free `CodeType` (a string with the `listID` attributes). The model does not bind a code list to a property: that would make it stricter than the standard, and reject values that a profile may legally use. Validating codes against a list is the job of a profile.

Tests

The tests run over every version, finding their fixtures by glob under `tests/fixtures/<version>/doc/<format>/<Document>/:`

- **XML**: the official OASIS examples of each version (only the ones of that version: each OASIS archive also carries the older ones). Each one is read, written, validated against the XSD and hopped through every format. UBL 2.0's signed example is a fixture of the first version.
- **JSON**: the official OASIS examples, read exactly as the note writes them, and everything the package writes is validated against the official JSON schemas. The XML and JSON examples of a document give the same data.

- **YAML:** this package's own: one minimal document per UBL document, commented with UBL's definitions. The 2.1 ones are reviewed by hand, the others are written by `generate-yaml-fixtures.php`.
- **Across versions:** the examples of an older version, read and written by the model of every later one, validate against its XSD.

The official `UBL-Invoice-2.0-Detached-Signature.xml` is the only official example that is not a fixture: it is a bare XML-DSig signature, not a UBL document.

Adding a UBL Version

1. Add it to `oasisVersions()` in `tools/oasis-common.php` and to `VersionRegistry::SUPPORTED`.
2. `php tools/generate-oasis.php <version>`, `php tools/patch-oasis.php <version>` and the formatter (or `composer run generate-oasis`).
3. `php tools/fetch-oasis-examples.php <version>` and `php tools/fetch-oasis-json.php <version>` (it skips the versions with no JSON note).
4. `php tools/generate-yaml-fixtures.php <version>`.
5. Run the tests: they pick the new version up without being told.

Last updated on 13/10/2026