

Archived - use

☰ OCDS 1.2 Upgrade Proposals we...

instead.

OCDS 1.2 Upgrade Proposals

Semantic issues

Planning vs contracting processes

Current status in OCDS 1.1.5

The standard and its documentation make frequent references to the “contracting process” defined in the Primer as:

“all the actions aimed at implementing one or more contracts. An individual contracting process has several different stages: **tendering, awarding, contracting and implementation.**”

While within the standard schema `Planning` is defined as:

“Information from the **planning phase of the contracting process.**”

The planning process is therefore treated as a part of the contracting process, **implying that the planning stage leads to a single tendering stage.**

What is the issue?

In many jurisdictions **a single planning process can lead to multiple separate contracting processes.** Some publishers currently publish planning details separately from tender, award and contract details for this reason, using the `relatedProcesses` array to link the planning process to any subsequent processes. However this is a work around, with OCDS 1.1.5 lacking any clear and consistent guidance on how to tackle this conceptual inconsistency. **As a core concept the definition of the contracting process should be clear and its relationship to planning unambiguous.**

What we are proposing

OCDS 1.1.5	OCDS 1.2
{	{

```

"ocid": "ocds-213czf-000-0001",
"tags": [
  "planning"
],
"planning": {
  "rationale": "The roads within
the city centre require
resurfacing."
}
}

```

Followed by a second publication with the same ocid containing the tender stage details:

```

{
  "ocid": "ocds-213czf-000-0001",
  "tags": [
    "tender"
  ],
  "tender": {
    "title": "Road resurfacing in
central Edinburgh."
  }
}

```

```

"ocid": "ocds-213czf-000-0001",
"tags": [
  "planning"
],
"planning": {
  "rationale": "The roads within
the city centre require
resurfacing."
}
}

```

Followed by a second publication with a new ocid that links to the previous publication using the `relatedProcesses` array:

```

{
  "ocid": "ocds-213czf-000-0002",
  "tags": [
    "tender"
  ],
  "tender": {
    "title": "Road resurfacing in
central Edinburgh."
  },
  "relatedProcesses": [
    {
      "id": "0001",
      "relationship": [
        "planning"
      ],
      "scheme": "ocid",
      "identifier":
"ocds-213czf-000-0002"
    }
  ]
}

```

For OCDS 1.2 any proposed change to the concepts used must be backwards compatible. We will add guidance encouraging publishers to publish details of a planning process under a separate ocid from the contracting process(es) that result from it, and reference this in the contracting process release(s) using `relatedProcesses`. The planning process release

should omit the `awards` and `contracts` arrays and the contracting process release(s) should omit the `planning` object.

In addition we shall clarify the definition of “Contracting process” to:

“All the actions aimed at implementing one or more contracts. This covers tendering, awarding, contracting and implementation. It does not include actions linked to planning, as these are often less structured and may be linked to multiple contracting processes. In multiple stage procedures (e.g. framework agreements with reopening of competition), each round of competition is treated as a separate contracting process.

Procedures that failed and were restarted are considered new processes.

Boundaries between processes (e.g. whether two contracts result from a single process or from two processes) are set by buyers depending on their needs (e.g. efficient division of labor, clear communication with the market) and legislation (e.g. rules on using procedures and lots).”

And the definition of “Planning process” to:

“All the actions aimed at planning one or more contracting processes. This covers, for example, establishing the rationale for the procurement, giving the market a general description of the purchase, getting the necessary budget, forecasting and conducting market research.

Planning processes are often less structured than contracting processes, so one or more planning processes may lead to one or more contracting processes.”

making it clear that the planning process is separate from the contracting process that results from it.

Github links

[planning vs contracting](#)

Stages of the contracting process

Current status in OCDS 1.1.5

OCDS 1.1.5 defines 4 top level stages in the contracting process. Each stage is represented by 3 different data items within the schema, a high level object/array, a subschema and a release tag. These are defined as follows:

Stage	High level object/array	Subschema	release `tag` code
planning	`planning` = "Information from the planning phase of the contracting process. This includes	`Planning` = "Information from the planning phase of the contracting process. Note that many other	`planning` = "A contracting process is proposed or planned. Information in the tender section describes the

	information related to the process of deciding what to contract, when and how."	fields can be filled in a planning release, in the appropriate fields in other schema sections; these would likely be estimates at this stage, e.g. value in tender."	proposed process. The tender.status field should be used to identify whether the planning is at an early pipeline stage, or whether there are detailed plans for a tender developed."
tender	`tender` = "The activities undertaken in order to enter into a contract."	`Tender` = "Data regarding tender process - publicly inviting prospective contractors to submit bids for evaluation and selecting a winner or winners."	`tender` = "Providing information about a new tender (call for proposals) process. Tender release should contain details of the goods or services being sought."
award	`awards` = "Information from the award phase of the contracting process. There can be more than one award per contracting process e.g. because the contract is split among different providers, or because it is a standing offer."	`Award` = "An award for the given procurement. There can be more than one award per contracting process e.g. because the contract is split among different providers, or because it is a standing offer."	`award` = "Providing information about the award of a contract. One or more award sections will be present, and the tender section might be populated with details of the process leading up to the award."
contract	`contracts` = "Information from the contract creation phase of the procurement process."	`Contract` = "Information regarding the signed contract between the buyer and supplier(s)."	`contract` = "Providing information about the details of a contract that has been, or will be, entered into. The tender section might be populated with details of the process leading up to the contract, and the award section might contain details of the award on the basis of which this contract will be signed."

What is the issue?

- Each stage has a slightly different description depending on the data item it is being referenced in, this introduces unnecessary ambiguity.

- None of the descriptions provide clarity over when the stage in question begins or ends. This makes it harder for publishers to work out where in OCDS they should publish their data.
- While `Award` and `Contract` correspond to real world objects with clear identities, `Tender` and `Planning` are OCDS concepts whose corresponding real world processes can be different depending on the jurisdiction. They require clear definitions in OCDS to help publishers be sure they are using them correctly.

What we are proposing

In OCDS 1.2 we shall expand and consolidate the definitions of the planning, tender, award and contract stages.

Stage	High level object/array	Subschema	release `tag` code
planning	`planning` = Information about, for example, needs identification, budget planning and market research. This information concerns the planning process. This information typically concerns the period before contracting documents (for example, procurement documents) are available to potential suppliers.	`Planning` = as `planning`	'planning' = as `planning`
tender	`tender` = Information about, for example, the needed items and their estimated value, procurement method, award criteria, and various deadlines. This information concerns either the contracting process or the planning process. For a contracting process, this information typically concerns the period starting with contracting documents (for example, procurement documents) being available to potential suppliers and ending with the bid submission deadline. For a planning process, this information typically concerns the period before contracting documents are	`Tender` = as `tender`	'tender' = as `tender`

	available to potential suppliers.		
award	<p>`awards` = Information about the awards. This information concerns the contracting process. This information typically concerns the period after the bid submission deadline and ending with the award or, if there is a standstill period, the end of the standstill period.</p>	<p>`Award` = Decision by the buyer or the procuring entity on the supplier with whom it intends to conclude a contract, including the items to be supplied and their value. Typically, this decision leads to one or more contracts, but not always (for example, the award is appealed at court or the supplier refuses to sign the contract). Depending on the jurisdiction, a single decision may concern a single supplier, item and value; or batches of suppliers and/or items and/or values. Similarly, sometimes the award is published as soon as it is made and sometimes only together with a contract.</p>	<p>‘award’ = as ‘awards’</p>
contract	<p>`contracts` = Information about the contracts and their implementation. This information concerns the contracting process. This information typically concerns the period after the award or, if there was a standstill period, after the end of the standstill period.</p>	<p>`Contract` = Information regarding the contract, typically between the buyer and supplier. This includes contracts describing all the contractual conditions (e.g. item, quantity, price, payment terms, time and place of delivery), as well as contracts only describing the general contractual conditions (such as a framework agreement) and those only describing the specific contractual conditions (such as a contract within a framework</p>	<p>‘contract’ = as ‘contract’</p>

		<p>agreement). Communication between contractual parties that consists of minor specifications of conditions agreed previously (e.g. specifying the time or place of delivery) is not considered a contract. Amendments are considered as part of the contract that is being amended. Contracts are also used when giving prizes or other rewards (e.g. a follow-up contract) resulting from a design contest.</p>	
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Github links

[update stage object descriptions and tags](#)
[contracts](#)
[award](#)

subheadings for the following issues under Semantic issues

-
- [buyer](#) vs [procuring entity](#)

Validations

New and deprecated fields and codes

Tax inclusive and exclusive values

Current status in OCDS 1.1.5

Monetary values are published using the `Value` subschema in all stages of the contracting process. This subschema contains two fields, `amount` defined as “amount as a number” and `currency`.

What is the issue?

Different jurisdictions have different rules on whether or not the published value of a procurement opportunity, an award or a contract needs to be tax inclusive or exclusive. In OCDS 1.1.5 there is no way to indicate within the data itself if the published values include tax or not.

What we are proposing

OCDS 1.1.5	OCDS 1.2
<pre>{ "tender": { "value": { "amount": 1000, "currency": "USD" } } }</pre>	<pre>{ "tender": { "value": { "amount": 1000, "amountGross": 1000, "amountNet": 800, "currency": "USD" } } }</pre>

We will add two new fields to the `Value` subschema, `amountGross` and `amountNet`. These can be used in place of `amount` or in addition as per the publisher's needs.

Github links

[value.amountNet and .amountGross](#)

Country field added, countryName deprecated

Current status in OCDS 1.1.5

The `address` subschema enables publishers to split a street address into its constituent components. This includes the country the address is located in, which is declared in the free text `countryName` field.

What is the issue?

Free text fields increase the potential for data entry mistakes in the data. Using a stable and well defined codelist would reduce the potential for such errors and help make geographic analysis of the data smoother.

What are we proposing?

OCDS 1.1.5	OCDS 1.2
<pre>{ "parties": [{ "id": "ORG-0001", "address": { "countryName": "Mexico" } }] }</pre>	<pre>{ "parties": [{ "id": "ORG-0001", "address": { "country": "MX" } }] }</pre>

We will deprecate the `countryName` field and replace it with a `country` field that will only accept codes from the new closed country.csv codelist. The codes in country.csv will be taken from the ISO 3166-1 alpha2 codelist.

Github links

[country field added, countryName deprecated](#)

subheadings for the following issues

- [status codelists](#) updates
- [deprecated funder](#)
-
-

Extensions

Current status in OCDS 1.1.5

OCP maintains multiple extensions to the OCDS that have been developed as and when a need has arisen. To make use of these a publisher must declare the extension(s) they are using as part of each release package they publish.

What is the issue?

Using the extension mechanism introduces an element of complexity to the publishing pipeline.

What are we proposing?

A number of these extensions are used routinely by a significant number of OCDS publishers, or they cover a concept that for OCDS 1.2 is now considered core. These extensions will be added to the core OCDS for version 1.2.

Location

The Location extension adds a `Location` object to describe geographic locations and the arrays `deliveryLocations` and `deliveryAddresses` to `tender` and `items` to record delivery locations and delivery addresses. It also adds a `description` field to the `Address` object. This is one of the most commonly used extensions with OCDS 1.1.5.

```
{
  "tender": {
    "deliveryAddresses": [
      {
        "streetAddress": "Town Hall, St Aldate's",
        "locality": "Oxford",
        "region": "Oxfordshire",
        "postalCode": "OX1 1BX",
        "country": "GB",
        "description": "The old town hall"
      }
    ],
    "deliveryLocations": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [
            51.751944,
            -1.257778
          ]
        },
        "gazetteer": {
          "scheme": "GEONAMES",
          "identifiers": [
            "2640729"
          ]
        },
        "description": "Central Oxford",
        "uri": "http://www.geonames.org/2640729/oxford.html"
      }
    ],
    "items": [
```

```
{
  "deliveryAddresses": [
    {
      Address object
    }
  ],
  "deliveryLocations": [
    {
      Location object
    }
  ]
}
]
```

Github links

<https://github.com/open-contracting/standard/issues/1484>

<https://github.com/open-contracting/standard/issues/1179>

Project

subheadings for each extension even though many will be very small sections but want to allow readers to just navigate straight to the extension they may be interested in.

- project, location, [submission terms \(partly\)](#), [legal basis \(classificationScheme codes\)](#), [organization classification](#) (classificationScheme codes), [tariffs \(documentType codes\)](#), EU ([partyRole](#) and [milestoneType](#) codes), PPP ([partyRole](#) and [milestone](#) codes) merged into core

Packaging

Others

New norms

[string identifiers preference](#)

[strengthening of normative statements in schema](#)

[release date uniqueness](#)