Release Reference

The *Release Schema* provides a detailed specification of the fields and data structures to use when publishing contracting data. Supplementary schemas show how to combine releases into data packages and how to compile releases into records.

This reference page provides key information on using the release schema.

Note: If any conflicts are found between this text, and the text within the schema, the schema takes precedence

Release structure

The majority of OCDS data is held within a release structure. One or more releases can be published within a release package. Releases are made up of a number of sections, including:

- planning
- tender
- award
- contract
- implementation

A release can only contain one tender section, but may contain multiple award, contract and implementation sections. This is because each OCDS release relates to *a single contracting process*, and we define contracting processes by their unique initiation (tender) stage.

Releases are given a tag to indicate the specific stage of a contracting process they represent. However, there are no formal restrictions on when information about a stage of the contracting process may be provided.

For example, a publisher announcing the signing of a contract with a 'contract' tag, may also include information in the award and tender blocks in order to provide a comprehensive picture of the contracting process to date which led to that contract being signed.

Package Metadata

Releases must be published within a *release package*, which can contain one or more releases. The release package, modelled on the Data Package protocol, provides meta-data about the release(s) it contains, the publisher, and data licensing information.

Field Name	Description	Format
uri	The URI of this package that identifies it uniquely in the world.	uri string
publishedDate	The date that this package was published. Ideally this should be the latest date that there is release information in this package.	date-time string
releases	See release-schema.json section for further details.	Object Array
publisher	Information to uniquely identify the publisher of this package. See publisher	Object
license	A link to the license that applies to the data in this package. A Public Domain Dedication or Open Definition Conformant license is strongly recommended. The canonical URI of the license should be used. Documents linked from this file may be under other license conditions.	uri string
publicationPolicy	A link to a document describing the publishers publication policy.	uri string

Notes:

- The uri should uniquely identify this release package. Publishers should provide a dereferenceable HTTP URI wherever possible and should host the data package at this URI, enabling users to look-up and verify the contents of a release package from its original source.
- The publishedDate on which this package was published. If a package is automatically generated and re-published on a regular basis, this date should reflect the date of the last change to the contents of the package.
- The publisher publisher block provides space for an organisation name and identifier.
- `license` - See the *licensing guidance* for more details on selecting and publishing license information.
- `publicationPolicy` - See the *publication policy* guidance for more details.

Release

The top level of a release consists of the following fields and objects:

Field Name	Description	Format

Field Name	Description	Format
ocid	A globally unique identifier for this Open Contracting Process. Composed of a publisher prefix and an identifier for the contracting process. For more information see the Open Contracting Identifier guidance	string
id	A unique identifier that identifies this release. A release ID must be unique within a release-package and must not contain the # character.	string
date	The date this information is released, it may well be the same as the parent publishedDate, it must not be later than the publishedDate from the parent package. It is used to determine merge order.	date-time string
tag	A value from the releaseTag codelist that identifies the nature of the release being made. Tags may be used to filter release, or, in future, for advanced validation when certain kinds of releases should contain certain fields.	Array
initiationType	String specifying the type of initiation process used for this contract, taken from the initiationType codelist. Currently only tender is supported.	string
planning	Information from the planning phase of the contracting process. This includes information related to the process of deciding what to contract for, when and how. See Planning	Object
tender	The activities undertaken in order to enter into a contract. See Tender	Object
buyer	The buyer is the entity whose budget will be used to purchase the goods. This may be different from the procuring agency who may be specified in the tender data. See Organization	Object
awards	Information from the award phase of the contracting process. There may be more than one award per contracting process e.g. because the contract is split amongst different providers, or because it is a standing offer. See Award section for further details.	Object Array
contracts	Information from the contract creation phase of the procurement process. See Contract section for further details.	Object Array

Field Name	Description	Format
language	Specifies the default language of the data using either two-digit ISO 639-1, or extended BCP47 language tags. The use of two-letter codes from ISO 639-1 is strongly recommended.	string

Notes:

- `ocid` - Providing each *contracting process* with a unique identifier is essential to enable data about contracts to be linked up across different releases. Open Contracting IDs are composed of a prefix assigned to each publisher, and a local identifier drawn from their internal systems that can be used to tie together tenders, awards, contracts and other key data points from a specific contracting process. See the *Open Contracting Identifier guidance* for details of how to construct an OCID.
- `tag` - The release tag is used to identify the nature of the release being made. This can be used by consuming applications to filter releases, or may in future be used for advanced validation. A release which updates or amends previous data must always use the appropriate update or amendment release tag. Values must be drawn from the releaseTag codelist.
- `date` - The release date should reflect the point in time at which the information in this release was disclosed. A release package may contain release with different release dates.
- `language` - see the section on multi-language support for information on language handling.
- `buyer` - The buyer details are published using an organization block.

Further details on each of the blocks contained within release are below.

Planning

The planning section can be used to describe the background to a contracting process. This may include details of the budget from which funds are drawn, or related projects for this contracting process. Background documents such as a needs assessment, feasibility study and project plan can also be included in this section.

Field Name	Description	Format
budget	See Budget	Object
rationale	The rationale for the procurement provided in free text. More detail can be provided in an attached document.	string

Field Name	Description	Format
documents	A list of documents related to the planning process. See Document section for further details.	Object Array

Apart from documents, the majority of information is held within the budget block. This is designed to allow both machine-readable linkable data about budgets, cross-referencing to data held in other standards such as the Fiscal Data Package or International Aid Transparency Initiative Standard, and human readable description of the related budgets and projects, supporting users to understand the relationship of the contracting process to existing projects and budgets even where linked data is not available.

Budget

Field Name	Description	Format
source	Used to point either to a corresponding Budget Data Package, or to a machine or human-readable source where users can find further information on the budget line item identifiers, or project identifiers, provided here.	uri string
id	An identifier for the budget line item which provides funds for this contracting process. This identifier should be possible to cross-reference against the provided data source.	string, integer
description	A short free text description of the budget source. May be used to provide the title of the budget line, or the programme used to fund this project.	string
amount	The value of the budget line item. See Value	Object
project	The name of the project that through which this contracting process is funded (if applicable). Some organizations maintain a registry of projects, and the data should use the name by which the project is known in that registry. No translation option is offered for this string, as translated values can be provided in third-party data, linked from the data source above.	string
projectID	An external identifier for the project that this contracting process forms part of, or is funded via (if applicable). Some organizations maintain a registry of projects, and the data should use the identifier from the relevant registry of projects.	string, integer
uri	A URI pointing directly to a machine-readable record about the related budget or projects for this contracting process.	uri string

Tender

The tender section includes details of the announcement that an organization intends to source some particular goods, works or services, and to establish one or more contract(s) for these.

It may contain details of a forthcoming process to receive and evaluate proposals to supply these goods and services, and may also be used to record details of a completed tender process, including details of bids received.

Field Name	Description	Format
id	An identifier for this tender process. This may be the same as the ocid, or may be drawn from an internally held identifier for this tender.	string, integer
title	Tender title	string
description	Tender description	string
status	The current status of the tender based on the tenderStatus codelist	string
items	The goods and services to be purchased, broken into line items wherever possible. Items should not be duplicated, but a quantity of 2 specified instead. See Item section for further details.	Object Array
minValue	The minimum estimated value of the procurement. See Value	Object
value	The total upper estimated value of the procurement. See Value	Object
procurementMethod	Specify tendering method against the method codelist as per GPA definitions of Open, Selective, Limited	string
procurementMethodDetails	Additional detail on the procurement method used. This field may be used to provide the local name of the particular procurement method used.	string
procurementMethodRationale	Rationale of procurement method, especially in the case of Limited tendering.	string
awardCriteria	Specify the award criteria for the procurement, using the award criteria codelist	string

Field Name	Description	Format
awardCriteriaDetails	Any detailed or further information on the award or selection criteria.	string
submissionMethod	Specify the method by which bids must be submitted, in person, written, or electronic auction. Using the submission method codelist	Array
submissionMethodDetails	Any detailed or further information on the submission method. This may include the address, e-mail address or online service to which bids should be submitted, and any special requirements to be followed for submissions.	string
tenderPeriod	The period when the tender is open for submissions. The end date is the closing date for tender submissions. See Period	Object
enquiryPeriod	The period during which enquiries may be made and answered. See Period	Object
hasEnquiries	A Yes/No field to indicate whether enquiries were part of tender process.	boolean
eligibilityCriteria	A description of any eligibility criteria for potential suppliers.	string
awardPeriod	The date or period on which an award is anticipated to be made. See Period	Object
numberOfTenderers	The number of entities who submit a tender.	integer
tenderers	All entities who submit a tender. See Organization section for further details.	Object Array
procuringEntity	The entity managing the procurement, which may be different from the buyer who is paying / using the items being procured. See Organization	Object

Field Name	Description	Format
documents	All documents and attachments related to the tender, including any notices. See the documentType codelist for details of potential documents to include. See Document section for further details.	Object Array
milestones	A list of milestones associated with the tender. See Milestone section for further details.	Object Array
amendment	See Amendment	Object

Notes:

- `tender.id` - see the identifiers guidance for further information on the tender identifier. In most cases this can be the same as the ocid.
- `procuringEntity` - in some cases the organization managing the procurement process may be different from the organization whose budget is being used for the procurement (the 'buyer' in OCDS terminology). If this is the case, then the details of this procuring organization should be provided here.
- `title` and `description` - tender title and description are optional. The details of items to be procured should always be provided in `items`. Descriptions should not be used in place of providing structured data on items, dates and other details. Instead, title and description should be used to provide a brief overview of the tender. Publishers should consider adopting a 'tweet length' title, and should avoid ALL UPPERCASE titles, or titles containing code words or other artefacts from internal databases. The goal of these fields is to give users a clear idea of the nature of a tender.
- `items` - publishers should provide details of each of the items to be procured under this tender.
- `milestones` - publishers should list any relevant milestones associated with the delivery of the goods and services covered by this tender. These are the milestones against which the whole contracting process will be evaluated. Publishers may include information about key milestones during the tender process itself, but should not use this in place of `tenderPeriod`, `enquiryPeriod` or `awardPeriod`.
- `value` and `minValue` - the total upper estimated value of a procurement should be given in `value`. For publishers who also specify a estimate minimum value, this can be placed in `minValue`.
- `procurementMethod` and `procurementMethodRationale`. Tendering

processes can use a variety of methods. Publishers should map their methods to one of the approved codes according to the GPA definitions of open, selective or limited. A free text explanation of why a given method was appropriate to this tender can be provided in `procurementMethodRationale`.
- `awardCriteria` and `awardCriteriaDetails` - The award criteria code list describes the basis on which contract awards will be made. This is an open codelist, and so may be extended with new codes. Free text describing the basis on which bids will be judged, and made, can be provided in the `awardCriteriaDetail` field. Publishers wishing to provide more structured information about selection, shortlisting and award criteria should propose *extensions* for this.
- `documents` - supporting documentation should be attached to the tender. This may include official legal notices of tender, as well as technical specifications, evaluation criteria, and, as a tender process progresses, clarifications, replies to queries and copies of bids submitted or listings of shortlisted firms. See the attachments section for more details of how to include documents, and consult the documentType codelist for suggested documents to include for basic, intermediate or advanced publication.

Information on bidders against a contract will be handled by an *extension* during the period of the standard release candidate. Publishers wishing to provide detailed information on bidders should *contact support*.

Award

The award section is used to announce any awards issued for this tender. There may be multiple awards made. Releases can contain all, or a subset, of these awards. A related award block is required for every contract, as it contains information on the suppliers.

Field Name	Description	Format
id	The identifier for this award. It must be unique and cannot change within the Open Contracting Process it is part of (defined by a single ocid). See the identifier guidance for further details.	string, integer
title	Award title	string
description	Award description	string
status	The current status of the award drawn from the awardStatus codelist	string
date	The date of the contract award. This is usually the date on which a decision to award was made.	date-time string

Field Name	Description	Format
value	The total value of this award. In the case of a framework contract this may be the total estimated lifetime value, or maximum value, of the agreement. There may be more than one award per procurement. See Value	Object
suppliers	The suppliers awarded this award. If different suppliers have been awarded different items of values, these should be split into separate award blocks. See Organization section for further details.	Object Array
items	The goods and services awarded in this award, broken into line items wherever possible. Items should not be duplicated, but the quantity specified instead. See Item section for further details.	Object Array
contractPeriod	The period for which the contract has been awarded. See Period	Object
documents	All documents and attachments related to the award, including any notices. See Document section for further details.	Object Array
amendment	See Amendment	Object

Contract

The contract section is used to provide details of contracts that have been entered into. Every contract needs to have a related award, linked via the `awardID` property. This is because supplier information is contained within the 'award'. The framework contract details below help illustrate the reasons for this.

Field Name	Description	Format
id	The identifier for this contract. It must be unique and cannot change within its Open Contracting Process (defined by a single ocid). See the identifier guidance for further details.	string, integer
awardID	The award.id against which this contract is being issued.	string, integer
title	Contract title	string
description	Contract description	string
status	The current status of the contract. Drawn from the contractStatus codelist	string
period	The start and end date for the contract. See Period	Object

Field Name	Description	Format
value	The total value of this contract. See Value	Object
items	The goods, services, and any intangible outcomes in this contract. Note: If the items are the same as the award do not repeat. See Item section for further details.	Object Array
dateSigned	The date the contract was signed. In the case of multiple signatures, the date of the last signature.	date-time string
documents	All documents and attachments related to the contract, including any notices. See Document section for further details.	Object Array
amendment	See Amendment	Object
implementation	Information related to the implementation of the contract in accordance with the obligations laid out therein. See Implementation	Object

Framework contracts

A framework is an agreement with suppliers to establish terms governing contracts that may be awarded during the life of the agreement*.

Framework **agreements** can be represented by awards, which would each detail a supplier participating in the framework, the line items that the agreement covers with this supplier, and any maximum value for the agreement over time.

Each call-off purchase against a framework agreement would result in a contract, related to the framework agreement award via `awardID`.

As a result, award and contract each contain an items block, allowing the award to describe the possible goods and services that can be supplied, whilst the contract describes those that are to be supplied in any particular instance.

Implementation

Implementation information can be updated over the course of a contract. It belongs nested within the contract it relates to. Implementation blocks include the following elements:

Field Name	Description	Format
transactions	A list of the spending transactions made against this contract See Transaction section for further details.	Object Array

Field Name	Description	Format
milestones	As milestones are completed, milestone completions should be documented. See Milestone section for further details.	Object Array
documents	Documents and reports that are part of the implementation phase e.g. audit and evaluation reports. See Document section for further details.	Object Array

Information on subcontracts is not currently included in the release candidate schema, but may be handled by *proposed extensions*

Transaction

Field Name	Description	Format
id	A unique identifier for this transaction. This identifier should be possible to cross-reference against the provided data source. For the budget data package this is the id, for IATI, the transaction reference.	string, integer
source	Used to point either to a corresponding Budget Data Package, IATI file, or machine or human-readable source where users can find further information on the budget line item identifiers, or project identifiers, provided here.	uri string
date	The date of the transaction	date-time string
amount	The value of the transaction. See Value	Object
providerOrganization	The Organization Identifier for the organization from which the funds in this transaction originate. Expressed following the Organizational Identifier standard - consult the documentation and the codelist. See Identifier	Object
receiverOrganization	The Organization Identifier for the organization which receives the funds in this transaction. Expressed following the Organizational Identifier standard - consult the documentation and the codelist. See Identifier	Object
uri	A URI pointing directly to a machine-readable record about this spending transaction.	uri string

The transaction block is modelled on the International Aid Transparency Initiative (IATI) transaction element, and can be used to represent actual flows of money between organisations in relation to this contract. As with the budget block, this may be used to cross-reference to a third party `source` of data, and can re-use identifiers from that source.

In most circumstances, the `providerOrganization` identifier should match that of the `buyer`, and the `recieverOrganization` identifier should match that of the `supplier`.

Milestones

See milestone reference below.

The implementation milestones should be updated to reflect when they are met.

Documents

Documents related to contract implementation should be stored here. This may include subcontracts.

See document reference below.

Amendment

A release may amend properties from a previous release. Whilst the release & record model of OCDS offers the opportunity to keep a full versioned history of changes, in many cases it is important for changes to a tender, award or contract to be explicitly declared.

The amendment block in each of tender, award and contract blocks provides the ability to explicitly declare changed fields, their former values, to provide an explanation for the change, and attach or reference relevant documentation.

Field Name	Description	Format
date	The data of this amendment.	date-time string
changes	An array change objects describing the fields changed, and their former values.	Array
rationale	An explanation for the amendment.	string

Amendment information should not be included in compiled records, as it is not possible to indicate **all** the former values of a field, only the most recent known value: whereas the version history of a full record can show all previous values.

Changes

Within each amendment block, publishers should provide an array of items that have changed, along with their former values.

Field Name	Description	Format
property	The property name that has been changed relative to the place the amendment is. For example if the contract value has changed, then the property under changes within the contract.amendment would be value.amount.	string
former_value	The previous value of the changed property, in whatever type the property is. See former_value	Object

Field reference

Organization

The organization block can be used to provide a legal identifier for an organization, and to give address and contact point information.

Field Name	Description	Format
identifier	The primary identifier for this organization. Identifiers that uniquely pick out a legal entity should be preferred. Consult the organization identifier guidance for the preferred scheme and identifier to use. See Identifier	Object
additionalIdentifiers	A list of additional / supplemental identifiers for the organization, using the organization identifier guidance. This could be used to provide an internally used identifier for this organization in addition to the primary legal entity identifier. See Identifier section for further details.	Object Array
name	The common name of the organization. The ID property provides an space for the formal legal name, and so this may either repeat that value, or could provide the common name by which this organization is known. This field could also include details of the department or sub-unit involved in this contracting process.	string

Field Name	Description	Format
address	See Address	Object
contactPoint	See ContactPoint	Object

Identifier

The identifier block provides a way to identify the legal entities involved in a contracting process. If a Contracting Process represents a contract arranged by the department or branch of a larger organization, the legal entity (usually the registered organization) should be described in the identifier section, with details of the branch or department given in the name, address and contact point as relevant.

Field Name	Description	Format
scheme	Organization identifiers be drawn from an existing identification scheme. This field is used to indicate the scheme or codelist in which the identifier will be found. This value should be drawn from the Organization Identifier Scheme.	string
id	The identifier of the organization in the selected scheme.	string, integer
legalName	The legally registered name of the organization.	string
uri	A URI to identify the organization, such as those provided by Open Corporates or some other relevant URI provider. This is not for listing the website of the organization: that can be done through the url field of the Organization contact point.	uri string

Address

Field Name	Description	Format
streetAddress	The street address. For example, 1600 Amphitheatre Pkwy.	string
locality	The locality. For example, Mountain View.	string
region	The region. For example, CA.	string
postalCode	The postal code. For example, 94043.	string
countryName	The country name. For example, United States.	string

ContactPoint

Field Name	Description	Format

Field Name	Description	Format
name	The name of the contact person, department, or contact point, for correspondence relating to this contracting process.	string
email	The e-mail address of the contact point/person.	string
telephone	The telephone number of the contact point/person. This should include the international dialling code.	string
faxNumber	The fax number of the contact point/person. This should include the international dialling code.	string
url	A web address for the contact point/person.	uri string

Document

Documents may be attached at a number of points within the standard: to planning, tenders, awards, contracts and milestones. Each document block can consist of multiple documents, classified using the documentType codelist.

The document block is also used to link to legal notices, which should have a documentType of 'notice'.

Field Name	Description	Format
id	A local, unique identifier for this document. This field is used to keep track of multiple revisions of a document through the compilation from release to record mechanism.	string, integer
documentType	A classification of the document described taken from the documentType codelist. Values from the provided codelist should be used wherever possible, though extended values can be provided if the codelist does not have a relevant code.	string
title	The document title.	string
description	A short description of the document. We recommend descriptions do not exceed 250 words. In the event the document is not accessible online, the description field can be used to describe arrangements for obtaining a copy of the document.	string
url	direct link to the document or attachment. The server providing access to this document should be configured to correctly report the document mime type.	uri string

Field Name	Description	Format
datePublished	The date on which the document was first published. This is particularly important for legally important documents such as notices of a tender.	date-time string
dateModified	Date that the document was last modified	date-time string
format	The format of the document taken from the IANA Media Types code list, with the addition of one extra value for 'offline/print', used when this document entry is being used to describe the offline publication of a document. Use values from the template column. Links to web pages should be tagged 'text/html'.	string
language	Specifies the language of the linked document using either two-digit ISO 639-1, or extended BCP47 language tags. The use of two-letter codes from ISO 639-1 is strongly recommended unless there is a clear user need for distinguishing the language subtype.	string

Date

OCDS makes use of ISO8601 date-times, following RFC3339 §5.6.

A time and timezone/offset MUST always be provided in a date-time.

The following are valid date-times:

- '2014-10-21T09:30:00Z' - 9:30 am on the 21st October 2014, UTC
- '2014-11-18T18:00:00-06:00' - 6pm on 18th November 2014 CST (Central Standard Time)

The following are not valid:

- '2014-10-21' - Missing time portion
- '2014-10-21T18:00' - missing seconds in time portion.
- '2014-11-18T18:00:00' - Missing timezone/offset portion
- '11/18/2014 18:00' - Not following the pattern at all!

Accurately including the time and timezone offsets is particular important for tender deadlines and other dates which may have legal significance, and where users of the data may be from different timezones. The character Z on the end of a date-time indicates the UTC (or Zero offset) timezone, whereas other timezones are indicated by their value '+/-hh:mm' UTC on the end of the date-time value.

In the event that the system from which data is drawn only includes dates, and does not include time information, publishers should consider sensible defaults for each field. For example, the startDate time of a clarification period may be set to '00:00:00Z' to indicate that clarifications can be requested from any time on the date stated, with the endDate time set to 23:59:59Z to indicate that clarifications can be sent up until the end of the endDate given. Alternatively, if clarification requests are only accepted in standard office hours, these values might be 09:00:00Z and 17:00:00Z respectively.

In the event that a date field is not bound to a specific time at all, publishers should choose a default time value of '23:59:59' and either 'Z' (for UTC) or the timezone of the publisher, indicating that the time refers to the end of the given date.

Period

A period is an object consisting of a start date and end date, represented as date-times.

Field Name	Description	Format
startDate	The start date for the period.	date-time string
endDate	The end date for the period.	date-time string

Item

The items block is used to list the line-items associated with a tender, award or contract.

Field Name	Description	Format
id	A local identifier to reference and merge the items by. Must be unique within a given array of items.	string, integer
description	A description of the goods, services to be provided.	string
classification	The primary classification for the item. See the itemClassificationScheme to identify preferred classification lists, including CPV and GSIN. See Classification	Object

Field Name	Description	Format
additionalClassifications	An array of additional classifications for the item. See the itemClassificationScheme codelist for common options to use in OCDS. This may also be used to present codes from an internal classification scheme. See Classification section for further details.	Object Array
quantity	The number of units required	integer
unit	Description of the unit which the good comes in e.g. hours, kilograms. Made up of a unit name, and the value of a single unit. See unit	Object

Notes:

- The *proposed location extension* can be attached to items, allowing the point of delivery for a given item or the site of works to be completed to be indicated in both the tender, award and contract stage.
- Items should be classified according to an established scheme of codes. A single primary `classification` can be given, although an array of `additionalClassification` can be provided.

Classification

Field Name	Description	Format
scheme	An classification should be drawn from an existing scheme or list of codes. This field is used to indicate the scheme/codelist from which the classification is drawn. For line item classifications, this value should represent an known Item Classification Scheme wherever possible.	string
id	The classification code drawn from the selected scheme.	string, integer
description	A textual description or title for the code.	string
uri	A URI to identify the code. In the event individual URIs are not available for items in the identifier scheme this value should be left blank.	uri string

Unit

The `unit` block allows detailed specification of the parameters and price of units that make up a line-item. Although no code list for units has been established in the current release of the standard, publishers may consider using the Units provided by the Quantities, Units, Dimensions and Data Types Ontologies in the `unit.name` field (drawing on the CamelCase unit names, such as SquareMile), in order to provide detailed information the cost per unit of a line-item.

Field Name	Description	Format
name	Name of the unit	string
value	The monetary value of a single unit. See Value	Object

Milestone

Milestone information can be included in the tender and contract implementation blocks.

In the context of a tender block, milestones describe the key deliverables of a contract, or key points during the lifetime of the contract. These may have associated documentation.

In the context of a contract implementation block, milestones are used to track progress towards those deliverables and key events set out in the tender.

Field Name	Description	Format
id	A local identifier for this milestone, unique within this block. This field is used to keep track of multiple revisions of a milestone through the compilation from release to record mechanism.	string, integer
title	Milestone title	string
description	A description of the milestone.	string
dueDate	The date the milestone is due.	date-time string
dateModified	The date the milestone was last reviewed or modified and the status was altered or confirmed to still be correct.	date-time string
status	The status that was realized on the date provided in dateModified, drawn from the milestoneStatus codelist.	string
documents	List of documents associated with this milestone. See Document section for further details.	Object Array

Notes:

- The `dateModified` field should be changed whenever the progress towards a milestone is reviewed, and the `status` either updated, or re-confirmed.

Value

Financial values should always be published with a currency attached.

Field Name	Description	Format
amount	Amount as a number.	number
currency	The currency in 3-letter ISO 4217 format.	string

During the period of the 1.0 RC, if information on taxation related to a value is required, this may be handled by an extension. A discussion of handling taxation can be found on GitHub.

Location

The addition of location information is currently handled through a *proposed extension* to the standard.

Publisher

The publisher block is used in release and record packages to identify the source of a dataset.

Field Name	Description	Format
name	The name of the organisation or department responsible for publishing this data.	string
scheme	The scheme that holds the unique identifiers used to identify the item being identified.	string
uid	The unique ID for this entity under the given ID scheme. Note the use of 'uid' rather than 'id'. See issue #245.	string
uri	A URI to identify the publisher.	uri string

Language

Many publishers need to be able to share key data in multiple languages. All free-text title and description fields in the Open Contracting Data Standard can be given in one or more languages.

Language variations are included by a copy of multi-lingual fields, suffixed with a language code.

E.g. `title` and `title_es`

In order to allow users to identify the language used in non-suffixed fields, OCDS release and records should declare the default language in the `language` field.

Languages should be identified using language tags taken from BCP47. The specification allows BCP47 values in order to accommodate variations in dialect where this is important. However, publishers **should** use the two letter ISO-639-1 two-digit language tags in the vast majority of circumstances, and should not assume that the users are able to distinguish between sub-tag variations (for example, OCDS publishers should strongly prefer 'en' over 'en_US' or 'en_GB').

To include a language variation of a field, the field name should be suffixed with _ and the appropriate language tag. For example: `title_es` for Spanish.

Worked example

A contract for 'Software consultancy services' may be published in a release with the default language sent to 'en' (the ISO-639-1 code for English). The following examples give the description of an item as English, French and Spanish.

json

```json
{
    "language": "en",
    "tender": {
        "item": {
            "description": "Software consultancy services",
            "description_es": "Servicios de consultoria en software",
            "description_fr": "Services de conseil en logiciels"
        }
    }
}
```

csv

description	description_es	dscription_fr
Software consultancy services	Servicios de consultoria en software	Services de conseil en logiciels

Release handling

The full OCDS data model is based on the idea of publishing a new release every time information about a contracting process changes. This way, users can gain a full view of change over time, and third-parties can understand what needs to be updated in any system that is tracking the progress of a contracting process.

Publishers will need to choose how to generate new releases, and whether to repeat information in each release, or just to provide changes. This choice should be based on an understanding the *merging process* that is used to generate a snapshot record of a full contracting process.

This model also requires publishers to pay careful attention to null values and missing fields.

Empty fields

Fields that are not being used, or that have no value, can be excluded in their entirety (key and value) from a published file.

Only including fields which have values will keep versioned datasets cleaner.

Emptying fields and values

There may be cases where a publisher needs to remove, rather than update, a value which was set in a previous release. In this case, the fields should explicitly be set to null.

The following describes how null values will be handled in the compilation of a record:

- If a field is included with a value in one release, and then set to `null` in a subsequent release, that field will be recorded as null in the compiled record (though past values of it should remain in the versioned section of the record).
- If a field is set to null in one release, and is also set to `null` in a subsequent release, it will be recorded as null as per the original release, and the record will not show any change based on the subsequent release I.e. fields that are set to null are recorded as such and then only changed if the value is set.
- If a field is set to `null` in one release, and then has a value in a subsequent release, it will first appear in the record, and the version section of the record as `null` and then with the subsequent value.
- If a field does not appear in one release, and then appears with a value in a

subsequent release, it will first appear in the record, and the version section of the record, when the first release that contains it is compiled into the record.