

ro-crate



Research Object Crate

[View the Project on GitHub](#) `ResearchObject/ro-crate`

---

## RO-Crate Metadata Specification 0.2

- Permalink: <https://w3id.org/ro/crate/0.2>
- Cite as: <https://doi.org/10.5281/zenodo.3406498> (this version)  
<https://doi.org/10.5281/zenodo.3406497> (any version)
- JSON-LD context: <https://w3id.org/ro/crate/0.2/context>
- Status: Working Draft
- Publisher: [researchobject.org](https://researchobject.org) community
- Published: 2019-09-12
- Editors: [Peter Sefton](#), [Eoghan Ó Carragáin](#), [Stian Soiland-Reyes](#)
- Authors: [Peter Sefton](#), [Eoghan Ó Carragáin](#), [Stian Soiland-Reyes](#), [Oscar Corcho](#), [Daniel Garijo](#), [Raul Palma](#), [Frederik Coppens](#), [Carole Goble](#), [José María Fernández](#), [Kyle Chard](#), [Jose Manuel Gomez-Perez](#), [Michael R Crusoe](#), [Ignacio Eguinoa](#), [Nick Juty](#), [Kristi Holmes](#), [Jason A. Clark](#), [Salvador Capella-Gutierrez](#), [Alasdair J. G. Gray](#), [Stuart Owen](#), [Alan R Williams](#)

1. Introduction & definition of an RO-Crate
  1. Terminology
  2. Linked data conventions

## 2. RO-Crate Structure

1. *RO-Crate Metadata File* (ro-crate-metadata.jsonld)
2. *RO-Crate Website* (ro-crate-preview.html and ro-crate-preview\_files/)
3. Payload files and directories
4. RO-Crates SHOULD be self-describing and self-contained

## 3. RO-Crate Metadata

1. RO-Crate uses linked data principles
2. Base metadata standard for RO-Crate: Schema.org
3. Additional metadata standards
4. Summary of Coverage
5. Recommended Identifiers
6. Core Metadata for the *Root Data Entity*
  1. Direct properties of the Root Data Entity
7. Contextual Entity: Person
  1. Contextual Entity: Organization
  2. Contextual Entity: Contact Point
8. Examples of referencing *Data Entities* (files and folders) from the *Root Data Entity*
  1. Example linking to a file and folders
  2. Adding detailed descriptions of encodings
9. Core Metadata for *Data Entities*
  1. Directory File Entity

## 4. Representing *Context Entities*

1. People
2. Organizations as values
3. More detail on ContactPoint
4. Publications
5. Publisher
6. Funding and grants
7. Licensing, Access control and copyright
8. Equipment
9. Software
10. Workflows and scripts
  1. Workflow diagram/sketch
11. Representing Provenance: changes to an object
12. Extra metadata such as Exif
13. Places
14. Time
15. Subjects & keywords
16. Digital Library and Repository content
17. Thumbnails

## 5. Extending RO-Crate

See <https://w3id.org/ro/crate> for further details about RO-Crate.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## Introduction & definition of an RO-Crate

*The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119.*

This document specifies a method, known as *RO-Crate* (Research Object Crate), of organizing file-based data with associated metadata, using linked data principles, in both human and machine readable formats, with the ability to include additional domain-specific metadata.

The core of RO-Crate is a JSON-LD file, the *RO-Crate Metadata File*, named `ro-crate-metadata.jsonld`. This file contains structured metadata about the dataset as a whole (the *Root Data Entity*) and, optionally, about some or all of its files. This provides a simple way to, for example, assert the authors (e.g. people, organizations) of the RO-Crate or one its files, or to capture more complex provenance for files, such as how they were created using software and equipment.

While providing the formal specification for RO-Crate, this document also aims to be a practical guide for software authors to create tools for generating and consuming research data packages, with explanation by examples.

### Terminology

*RO-Crate*: A directory structure that contains a dataset, which is described in an *RO-Crate Metadata File*.

*RO-Crate Root*: The top-level directory of the *RO-Crate*, indicated by the presence of the *RO-Crate Metadata File* `ro-crate-metadata.jsonld`

*RO-Crate Metadata File*: A JSON-LD file stored as `ro-crate-metadata.jsonld` in the *RO-Crate Root*. The metadata file describes the *RO-Crate* with structured data in form of *RO-Crate JSON-LD*.

*RO-Crate Website*: Human-readable HTML pages which describe the RO-Crate (i.e. the *Root Data Entity*, its *Data Entities* and *Context Entities*), with a home-page at `ro-crate-preview.html` (any additional files reside in `ro-crate-preview_files/`)

*Data Entity*: A JSON-LD representation, in the *RO-Crate Metadata File*, of a directory, file or other resource contained or described by the RO-Crate.

*Root Data Entity*: A *Data Entity* of type [Dataset](#), representing the RO-Crate as a whole.

*RO-Crate Metadata File Descriptor*: A *Contextual Entity* of type [CreativeWork](#), which describes the *RO-Crate Metadata File* and links it to the *Root Data Entity*.

*JSON-LD*: A JSON-based file format for storing *Linked Data*. This document assumes [JSON-LD 1.0](#). JSON-LD use a *context* to map from JSON keys to *URIs*.

*JSON*: The *JavaScript Object Notation (JSON) Data Interchange Format* as defined by [RFC7159](#); a structured text file format that can be programmatically consumed and generated in a wide range of programming languages.

*Context Entity*: A JSON-LD representation of an entity associated with a *Data Entity*, needed to adequately describe that *Data Entity*. For example, a *Person*, *Organization* (including research projects), *item of equipment (IndividualProduct)*, *license* or any other *thing* or *event* that forms part of the metadata for a *Data Entity* or supporting information.

*Linked Data*: A data structure where properties, types and resources are identified with *URIs*, which if retrieved over the Web, further describe or provide the identified property/type/resource.

*URI*: A *Uniform Resource Identifier* as defined in [RFC3986](#), for example `http://example.com/path/file.html` - commonly known as *URL*. In this document the term *URI* includes *IRI*, which also permit international Unicode characters.

*URI Path*: The relative *path* element of an *URI* as defined in [RFC3986 section 3.3](#), e.g. `path/file.html`

*RO-Crate JSON-LD Context*: A JSON-LD [context](#) that provides Linked Data mapping for RO-Crate metadata to vocabularies like [schema.org](#).

*RO-Crate JSON-LD*: JSON-LD using the *RO-Crate JSON-LD Context* containing RO-Crate metadata, which has been [flattened](#) and then [compacted](#) according to the rules in JSON-LD 1.0. The *RO-Crate JSON-LD* for an *RO-Crate* is stored in the *RO-Crate Metadata File*.

## Linked data conventions

Throughout this specification, RDF terms are referred to using the keys defined in RO-Crate JSON-LD Context.

Following [schema.org](#) practice, property names start with lowercase letters and Class names start with uppercase letters.

In `ro-crate-metadata.jsonld` RDF terms use their RO-Crate JSON-LD names as defined in the RO-Crate JSON-LD Context, which is available at <https://w3id.org/ro/crate/0.2/context>

## RO-Crate Structure

The structure an *RO-Crate* MUST follow is:

```
<RO-Crate root directory>/
| ro-crate-metadata.jsonld # _RO-Crate Metadata File_ MUST be present
| ro-crate-preview.html    # _RO-Crate Website_ homepage MAY be present
| ro-crate-preview_files/  # MAY be present
| | [_RO-Crate Website_ files]
| [payload files and directories] # 1 or more SHOULD be present
```

The name of the *RO-Crate root* directory is not defined. For instance, if an *RO-Crate* is archived in a ZIP-file, the *RO-Crate root* directory will correspond to the ZIP root directory.

### **RO-Crate Metadata File (ro-crate-metadata.jsonld)**

- The *RO-Crate Metadata File* MUST be named `ro-crate-metadata.jsonld` and appear in the *RO-Crate Root*
- `ro-crate-metadata.jsonld` MUST be a JSON-LD document which has been flattened and then compacted according to the rules in JSON-LD 1.1

## **RO-Crate Website (`ro-crate-preview.html` and `ro-crate-preview_files/`)**

In addition to the machine-oriented *RO-Crate Metadata File*, the RO-Crate MAY include a human-readable HTML rendering of the same information, known as the *RO-Crate Website*.

If present, `ro-crate-preview.html` MUST:

- Be a valid [HTML 5](#) document
- Contain at least a human readable summary of metadata relating to the *Root Data Entity*
- Contain a copy of the *RO-Crate JSON-LD* in a `script` element of the head element of the HTML, for example:

```
<script type="application/ld+json">
{
"@context": "https://w3id.org/ro/crate/0.2/context",
"@graph": [ ... ]
}
</script>
```

`ro-crate-preview.html` SHOULD:

- Contain at least the same information as the *RO-Crate JSON-LD*, with the exception that files which have no description, author or similar metadata MAY not be listed in the website.
- Display at least the metadata relating to the *Root Data Entity* as static HTML without the need for scripting. It MAY contain extra features enabled by JavaScript.
- When a *Data Entity* or *Context Entity* is referenced by its ID:
  - If it has a `name` property, provide a link to its HTML version.
  - If it does not have a name (e.g. a [GeoCoordinates](#) location), show it embedded in the HTML for the entity.
- For keys that resolve in the RO-Crate JSON-LD Context to a URI, indicate this (the simplest way is to link the key to its definition).
- For external URI values, provide a link.
- If there is sufficient metadata, contain a prominent “*Cite-as*” text with a natural language data citation (see for example the [FORCE11 Data Citation Principles](#)).
- If there are additional resources necessary to render the preview (e.g. CSS, JSON, HTML), link to them in a subdirectory `ro-crate-preview-files/`

## **Payload files and directories**

These are the actual files and directories that make up the dataset being described.

The base RO-Crate specification makes no assumptions about the presence of any specific files or folders beyond the reserved RO-Crate files described above. Payload files may appear directly in the *RO-Crate Root* alongside the *RO-Crate Metadata File*, and/or appear in sub-directories of the RO-Crate Root. Each file and directory MAY be represented as Data Entities in the *RO-Crate Metadata File*.

## **RO-Crates SHOULD be self-describing and self-contained**

A minimal RO-Crate is a directory containing a single *RO-Crate Metadata File*.

At the basic level, an RO-Crate is a collection of files represented as a schema.org *Dataset*, that together form a meaningful unit for the purposes of communication, citation, distribution, preservation, etc. The *RO-Crate Metadata File* describes the RO-Crate, and MUST be stored in the *RO-Crate Root*. Self-containment is a core principle of RO-Crate, i.e. that all Dataset files and relevant metadata SHOULD, as far as possible, be contained by the RO-Crate, rather than referring to external resources. However the RO-Crate MAY also reference external resources which are stored or accessed separately, via URIs, e.g. because these cannot be included for practical or legal reasons.

It is important to note that the *RO-Crate Metadata File* is not an exhaustive manifest or inventory, that is, it does not necessarily list or describe all files in the package. Rather it is focused on providing sufficient amount of metadata to understand and use the content and is designed to be compatible with existing and future approaches that *do* have full inventories / manifest and integrity checks, e.g. by using checksums, such as [BagIt](#) and Oxford Common File Layout [OCFL] [Objects](#).

The intention is that RO-Crates can work well with a variety of archive file formats, e.g. tar, zip, etc., and approaches to capturing file manifests and file fixity, such as [BagIt](#) and [OCFL](#), [git].

## RO-Crate Metadata

### RO-Crate uses linked data principles

**TODO** RO-Crate uses Linked Data principles [TODO - ref] - using URIs as names for things.

### Base metadata standard for RO-Crate: Schema.org

[Schema.org](#) is the base metadata standard for RO-Crate. Schema.org was chosen because it is widely used on the World Wide Web and supported by search engines, on the assumption that discovery is likely to be maximized if search engines index the content. NOTE: As far as we know there is no alternative, well-maintained linked-data schema for research data with the coverage needed for this project - i.e. a single standard for expressing all the examples presented in this specification.

RO-Crate relies heavily on [schema.org](#) using a constrained subset of [JSON-LD](#), and this document gives opinionated recommendations on how to represent the metadata using existing linked data best practices

Generally, the standard keys for [schema.org](#) should be used. However, RO-Crate uses variant names for some elements, specifically:

- `File` is mapped to <http://schema.org/MediaObject> which was chosen as a compromise as it has many of the properties that are needed to describe a generic file. Future versions of schema.org or a research data extension may re-define `File`.
- `Journal` is mapped to <http://schema.org/Periodical>.

### Additional metadata standards

The following terms from the from the [Research Object ontologies](#) are used:

- `Workflow` for a workflow definition, mapped to <http://purl.org/wf4ever/wfdesc#Workflow>
- `WorkflowScript` for a script, mapped to <http://purl.org/wf4ever/wf4ever#Script>
- `WorkflowSketch` for an image depicting a workflow, mapped to <http://purl.org/wf4ever/roterms#Sketch>.

RO-Crate also uses the *Portland Common Data Model* (PCDM) and imports these terms:

- RepositoryObject mapped to <https://pcdm.org/2016/04/18/models#Object>
- RepositoryCollection mapped to <https://pcdm.org/2016/04/18/models#Collection>
- RepositoryFile mapped to <https://pcdm.org/2016/04/18/models#Collection>
- hasMember mapped to <https://pcdm.org/2016/04/18/models#hasMember>
- hasFile mapped to <https://pcdm.org/2016/04/18/models#hasFile>

The keys RepositoryObject and RepositoryCollection were chosen to avoid collision between the terms Collection and Object with other vocabularies.

## Summary of Coverage

RO-Crate is simply a way to make metadata assertions about a set of files and folders that make up a *Dataset*. These assertions can be made at three levels:

- Assertions at the RO-Crate/*Dataset*\_level: for an RO-Crate to be useful, some metadata should be provided about the dataset as a whole (see minimum requirements for different use-cases below). In the *\_RO-Crate Metadata File*, we distinguish the *Root Data Entity* which represents the RO-Crate as a whole, from other *Data Entities* (files and folders contained in the RO-Crate) and *Context Entities*, e.g. a person, organisation, place related to an RO-Crate *Data Entity*
- Assertions about files and folders contained in the RO-Crate: in addition to providing metadata about the RO-Crate as a whole, RO-Crate allows metadata assertions to be made about any other *Data Entity*
- Variable-level assertions: In some cases, e.g. for tabular data, additional metadata may be provided about the structure and variables within a given file.

This document has guidelines for ways to represent common requirements for describing data in a research context, e.g.:

- Contact information for a data set.
- Descriptive information for a dataset and the files within it and their contexts such as an abstract, spatial and temporal coverage.
- Associated Publications.
- Funding relationships.
- Provenance information of various kinds; who (people and organizations) and what (instruments and computer programs) created or contributed to the data set and individual files within it.
- Workflows that operate on the data using standard workflow descriptions including 'single step workflows'; executable files or environments such as singularity containers or Jupyter notebooks.

However, as RO-Crate uses *Linked Data* principles, adopters of RO-Crate are free to supplement RO-Crate using [schema.org](https://schema.org) metadata and/or assertions using other *Linked Data* vocabularies.

## Recommended Identifiers

*RO-Crate JSON-LD* SHOULD use the following IDs where possible:

- For a *Root Data Entity*, an identifier which SHOULD be a DOI URL.
- For People participating in the research process: ORCID identifiers.
- For Organizations including funders, Research Organization Registry URIs. <https://ror.org/>
- For items of type Place, a geonames URL.

- For file formats, a [Pronom URL](https://www.nationalarchives.gov.uk/PRONOM/fmt/831), for example <https://www.nationalarchives.gov.uk/PRONOM/fmt/831>.

In the absence of the above, RO-Crates SHOULD contain stable persistent URIs to identify all entities wherever possible.

### Core Metadata for the *Root Data Entity*

The *RO-Crate JSON-LD* MUST contain a *Root Data Entity*, identified by a *RO-Crate Metadata File Descriptor* of type [CreativeWork](#) which describes it, with an [about](#) property referencing the *Root Data Entity* the *Root Data Entity* MUST have an `@id` of `./`.

```
{
  "@type": "CreativeWork",
  "@id": "ro-crate-metadata.jsonld",
  "identifier": "ro-crate-metadata.jsonld",
  "about": {"@id": "./"}
},

{
  "@id": "./",
  "@type": "Dataset",
  ...
}
```

To ensure a base-line interoperability between RO-Crates, and for an RO-Crate to be considered a *Valid RO-Crate*, a minimum set of metadata is required for the *Root Data Entity*. As stated above the *RO-Crate Metadata File* is not an exhaustive manifest or inventory, that is, it does not necessarily list or describe all files in the package. For this reason, there are no minimum metadata requirements in terms of describing *Data Entities* (files and folders) other than the *Root Data Entity*. Extensions of RO-Crate dealing with specific types of dataset may put further constraints or requirements of metadata beyond the *Root Data Entity* (see *Extending RO-Crate* below).

The *RO-Crate Metadata File Descriptor* MAY contain information such as licensing for the *RO-Crate Metadata File* so metadata can be licensed separately from Data.

The table below outlines the properties that the *Root Data Entity* MUST have to be minimally valid and additionally highlights properties required to meet other common use-cases, including the minimum metadata necessary to mint a DataCite DOI:

### Direct properties of the *Root Data Entity*

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	Institutional data repository (JISC RDSS)	Data discovery (Google Dataset Search)
@type	Must be a schema:Dataset	M	M	M	M



schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	Institutional data repository (JISC RDSS)	Data discovery (Google Dataset Search)
@id	Must be a a string of './'	M	M	M	M
name		M	M	M	M
datePublished		M	M	M	
author	Must appear at least once	referencing a Contextual Entity of @type: Person or Organization	?	M	M
license		M	M	M	
identifier		?	M	M	
distribution					?
contactPoint	Must appear at least once	referencing a Contextual Entity of @type ContactPoint			
publisher		?	M		
description		?			M
url	Must be a valid <i>URI</i> or <i>URI Path</i>		M		
hasPart	Must appear for each additional Data Entity described in the RO-Crate JSON-LD which is a direct child of the Root Data Entity				

### Contextual Entity: Person

(required since *author* is required)

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
@type		Y	M		
@id		Y	M	M	

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
name		Y	M		
familyName			O	M	
givenName			O	M	
identifier		?	O	O	
sameAs					R

### Contextual Entity: Organization

(required since author is required)

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
@type		Y		Y	
@id		Y		Y	
name		Y		Y	

### Contextual Entity: Contact Point

(required if contactPoint is required)

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
@type		?			?
@id		?			?
name		?			?
email		?			?

The following *RO-Crate Metadata File* represents a minimal description of an *RO-Crate*. Note that for brevity the *RO-Crate JSON-LD Context* is here shown by reference rather than inline.

```
{ "@context": "https://w3id.org/ro/crate/0.2/context",
  "@graph": [

  {
    "@id": "./",
    "identifier": "https://doi.org/10.4225/59/59672c09f4a4b",
    "@type": "Dataset",
    "datePublished": "2017",
    "name": "Data files associated with the manuscript:Effects of facilitated
```

```

      "distribution": { "@id": "https://data.research.uts.edu.au/examples/v1.0/"
    },
    "contactPoint": {
      "@id": "tim.luckett@uts.edu.au"
    }
  },
  {
    "@id": "https://data.research.uts.edu.au/examples/v1.0/timluckett.zip",
    "contentUrl": "https://data.research.uts.edu.au/examples/v1.0/timluckett.",
    "@type": "DataDownload",
    "encodingFormat": "application/zip"
  },
  {
    "@id": "tim.luckett@uts.edu.au",
    "@type": "ContactPoint",
    "contactType": "customer service",
    "email": "tim.luckett@uts.edu.au",
    "identifier": "tim.luckett@uts.edu.au",
    "url": "https://orcid.org/0000-0001-6121-5409"
  }
]
}

```

## Examples of referencing *Data Entities* (files and folders) from the *Root Data Entity*

Where files and folders are represented as *Data Entities* in the RO-Crate JSON-LD, these MUST be linked to, either directly or indirectly, from the Root Data Entity using the `hasPart` property. Directory hierarchies MAY be represented with nested Dataset *Data Entities*, or the Root Dataset MAY refer to files anywhere in the hierarchy using `hasPart`.

*Data Entities* representing files MUST be of `@type: File`, which is an RO-Crate alias for <http://schema.org/MediaObject>

*Data Entities* representing directories MUST be of `@type: Dataset`.

*Note: as indicated above, there is no requirement to represent every file and folder in an RO-Crate as Data Entities in the RO-Crate JSON-LD.*

### Example linking to a file and folders

```

<base directory>/
| ro-crate-metadata.jsonld
| cp7glop.ai
| lots_of_little_files/
| | file1
| | file2
| | ...
| | file54

```

An example *RO-Crate JSON-LD* for the above would be as follows:

```
{ "@context": "https://w3id.org/ro/crate/0.2/context",
  "@graph": [
    {
      "@id": "./",
      "@type": [
        "Dataset"
      ],
      "hasPart": [
        {
          "@id": "./cp7glop.ai"
        },
        {
          "@id": "./lots_of_little_files"
        }
      ],
    },
    {
      "@id": "./cp7glop.ai",
      "@type": "File",
      "contentSize": "383766",
      "description": "Illustrator file for Glop Pot",
      "encodingFormat": "application/pdf"
    },
    {
      "@id": "./lots_of_little_files",
      "@type": "Dataset",
      "description": "This directory contains many small files, that we're not sure we can read",
      "name": "Too many files",
    }
  ]
}
```

### Adding detailed descriptions of encodings

The above example provides a mime-type for the file `./cp7glop.ai` - which is useful as it may not be apparent that the file readable as a PDF file from the extension. To add more detail, encodings SHOULD be linked using a [PRONOM](#) identifier to a *Contextual Entity* of `@type Website`.

```
{
  "@id": "cp7glop.ai",
  "@type": "File",
  "contentSize": "383766",
  "description": "Illustrator file for Glop Pot",
  "encodingFormat": ["application/pdf", {"@id": "https://www.nationalarchives.gov.uk/PRONOM/fmt/19"}],
  {
    "@id": "https://www.nationalarchives.gov.uk/PRONOM/fmt/19",
    "name": "Acrobat PDF 1.5 - Portable Document Format",
  }
}
```

```
"@type": "Website"
}
```

If there is not PRONOM identifier, then a contextual entity with a URL as an @id MAY be used:

For example:

```
{
  "@id": "1st-tool.cwl",
  "@type": "File",
  "contentSize": "120",
  "description": "An example Common Workflow Language File",
  "encodingFormat": ["text/plain", {"@id": "https://www.commonwl.org/v1.0/W"},
},
{
  "@id": "https://www.commonwl.org/v1.0/Workflow.html",
  "name": "Common Workflow Language (CWL) Workflow Description, v1.0.2",
  "@type": "Website"
}
```

If there is no web-accessible description for a file format it SHOULD be described locally in the dataset, for example in a file:

```
{
  "@id": "some-file.some_extension",
  "@type": "File",
  "contentSize": "120",
  "description": "A file in a non-standard format",
  "encodingFormat": ["text/plain", {"@id": "https://www.commonwl.org/v1.0/W"},
},
{
  "@id": "some_extension.md",
  "encodingFormat": "text/plain",
  "name": "Description of some_extension file format",
  "@type": ["File", "CreativeWork"]
}
```

## Core Metadata for *Data Entities*

The table below outlines the properties that Data Entities, when present, MUST have to be minimally valid.

#### File Data Entity

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
-----------------	-------------	----------------	------------------------------	-----------	--

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
@type	MUST be a File (alias of MediaObject)	Y		Y	
@id	MUST be a <i>URI Path</i> relative to the <i>RO Crate root</i>	Y		Y	
name		Y		Y	
contentSize		?		Y	
dateModified		?		Y	
fileFormat		?			
encodingFormat		?			

### Directory File Entity

schema property	constraints	Valid RO-Crate	Citation Use-case (DataCite)	JISC RDSS	Data discovery (Google Dataset Search)
@type	MUST be Dataset				

## Representing *Context Entities*

The *RO-Crate JSON-LD @graph* SHOULD contain additional information about *Context Entities* for the use of both humans (in `ro-crate-preview.html`) and machines (in `ro-crate-metadata.jsonld`). This also helps to maximise the extent to which an *RO-Crate* is self-contained and self-describing, in that it reduces the need for the consumer of an *RO-Crate* to refer to external information which may change or become unavailable over time.

### People

A core principle of Linked data is to use URIs to identify things such as people. The following is the minimum recommended way of representing a `author` in a *RO-Crate*. This property MAY be applied in the context of a directory ([Dataset](#)) or to a [File](#).

```
{
  "@type": "Dataset",
  "@id": "some_id",
  "author": {"@id": "https://orcid.org/0000-0002-8367-6908"}
}
{
  "@id": "https://orcid.org/0000-0002-8367-6908",
  "@type": "Person",
  "affiliation": "University of Technology Sydney",
```

```
"name": "J. Xuan"  
}
```

This uses an [ORCID](#) to unambiguously identify an author, with a *Contextual Entity* of type [Person](#).

Note the string-value of the organizational affiliation. This SHOULD be improved by also providing a *Context Entity* for the organization (see example below).

## Organizations as values

An [Organization](#) SHOULD be the value for the [publisher](#) property of a [Dataset](#) or [ScholarlyArticle](#) or [affiliation](#) property of a [Person](#).

```
{  
  "@type": "Dataset",  
  "@id": "some_id",  
  "publisher": {"@id": "https://ror.org/03f0f6041"}  
}  
  
{  
  "@id": "https://ror.org/03f0f6041",  
  "@type": "Organization",  
  "name": "University of Technology Sydney",  
  "url": "https://ror.org/03f0f6041"  
}
```

An [Organization](#) SHOULD also be used for a [Person](#)'s affiliation.

```
{  
  "@type": "Dataset",  
  "@id": "some_id",  
  "publisher": {"@id": "https://ror.org/03f0f6041"},  
  "author": {"@id": "https://orcid.org/0000-0002-3545-944X"}  
},  
{  
  "@id": "https://ror.org/03f0f6041",  
  "@type": "Organization",  
  "name": "University of Technology Sydney"  
},  
{  
  "@id": "https://orcid.org/0000-0002-3545-944X",  
  "@type": "Person",  
  "affiliation": {"@id": "https://ror.org/03f0f6041"},  
  "email": "peter.sefton@uts.edu.au",  
  "name": "Peter Sefton"  
}
```

Sometimes researchers want to affiliate with a sub-part of an institution, such as institute, faculty department, or research group.

Here's a (real, but abridged) example of a researcher who has four institutional affiliations for a published article and data set:

```
{
  "@id": "./",
  "identifier": "https://doi.org/10.4225/59/59672c09f4a4b",
  "@type": "Dataset",
  "author": {"@id": "https://orcid.org/0000-0002-6756-6119"}
},
{
  "@id": "https://orcid.org/0000-0002-6756-6119",
  "@type": "Person",
  "affiliation": [
    {
      "@id": "1"
    },
    {
      "@id": "2"
    },
    {
      "@id": "3"
    },
    {
      "@id": "4"
    }
  ],
  "name": "Meera Agar"
},
```

The first affiliation is a Faculty of a university. The Faculty is associated with the university via [memberOf](#).

```
{
  "@id": "1",
  "@type": "Organization",
  "memberOf": {
    "@id": "https://ror.org/03f0f6041"
  },
  "name": "Faculty of Health, University of Technology Sydney"
},
{
  "@id": "https://ror.org/03f0f6041",
  "@type": "Organization",
  "name": "University of Technology Sydney"
},
```

Thus *RO-Crate JSON-LD* MAY express chained affiliations with more precision than by using string-literals as values for [affiliation](#).

[See an example of a Person with four affiliations.](#)



## More detail on ContactPoint

A RO-Crate SHOULD have contact information, using a [contactPoint](#) property with a value of [ContactPoint](#). Note that [Dataset](#) does not (yet) have a contactPoint property, so strictly this would have to be on a Person or Organization which is related to the Dataset via a [author](#) or [contributor](#) property.

NOTE: This usage follows that of Google Dataset search.

IMPLEMENTATION NOTE: The Google structured Data tool gives an error if `mailto:` URLs are used in the "url" property.

```
{
  "@id": "https://orcid.org/0000-0001-6121-5409",
  "@type": "Person",
  "affiliation": {
    "@id": "1"
  },
  "contactPoint": {
    "@id": "tim.luckett@uts.edu.au"
  },
  "familyName": "Luckett",
  "givenName": "Tim",
  "identifier": "https://orcid.org/0000-0001-6121-5409",
  "name": "Tim Luckett"
},

{
  "@id": "tim.luckett@uts.edu.au",
  "@type": "ContactPoint",
  "contactType": "customer service",
  "email": "tim.luckett@uts.edu.au",
  "identifier": "tim.luckett@uts.edu.au",
  "url": "https://orcid.org/0000-0001-6121-5409"
},
```

This is not ideal, as there is no direct semantic relationship between the contactPoint property and the Dataset, so the same [contactPoint](#) should be added to the Root Data Entity, in anticipation of this being added to Schema.org, even though this is not strictly supported at the moment.

```
{
  "@id": "./",
  "identifier": "https://doi.org/10.4225/59/59672c09f4a4b",
  "@type": "Dataset",
  "contactPoint": {
    "@id": "tim.luckett@uts.edu.au"
  },
}
```

See an example of an item of @type ContactPoint, which is linked to a Dataset and to an Organization via a contactPoint property.

## Publications

To associate a publication with a dataset the *RO-Crate JSON-LD* MUST include a URL (for example a DOI URL) as the @id of a publication using the `citation` property.

For example:

```
"citation": {"@id": "https://doi.org/10.1109/TCYB.2014.2386282"}
```

The publication SHOULD be described in the *RO-Crate JSON-LD*.

```
{
  "@id": "https://doi.org/10.1109/TCYB.2014.2386282",
  "@type": "ScholarlyArticle",
  "author": [
    {
      "@id": "https://orcid.org/0000-0002-8367-6908"
    },
    {
      "@id": "https://orcid.org/0000-0003-0690-4732"
    },
    {
      "@id": "https://orcid.org/0000-0003-3960-0583"
    },
    {
      "@id": "https://orcid.org/0000-0002-6953-3986"
    }
  ],
  "identifier": "https://doi.org/10.1109/TCYB.2014.2386282",
  "issn": "2168-2267",
  "name": "Topic Model for Graph Mining",
  "journal": "IEEE Transactions on Cybernetics",
  "datePublished": "2015"
}
```

See an example of a journal article.

## Publisher

The *Root Data Entity* SHOULD have a `publisher` property. This SHOULD be an `Organization` though it MAY be a `Person`, a string-literal, or a URI.

```
{
  "@id": "https://doi.org/10.5281/zenodo.1009240",
  "@type": "Dataset",
  "name": "Sample dataset for RO-Crate v0.2",
  "publisher": {
    "@id": "https://ror.org/03f0f6041"
  },
}
```

```

    "temporalCoverage": "2017"
  },
  {
    "@id": "https://ror.org/03f0f6041",
    "@type": "Organization",
    "identifier": "https://ror.org/03f0f6041",
    "name": "University of Technology Sydney"
  },

```

See an example of an Organization which is linked to the publisher property of a Dataset.

## Funding and grants

To associate a research project with a [Dataset](#), the *RO-Crate JSON-LD* SHOULD contain an entity for the project using type "Organization", referenced by a [funder](#) property. The project organization SHOULD in turn reference any external [funder](#), either by using its URL as an `@id` or via a *Context Entity* describing the funder.

NOTE: To make it very clear where funding is coming from, the *Root Data Entity* SHOULD also reference funders directly, as well as via a chain of references.

```

{
  "@id": "https://doi.org/10.5281/zenodo.1009240",
  "@type": "Dataset",
  "funder": {
    "@id": "https://ror.org/038sjwq14"
  },
},
{
  "@id": "https://ereseach.uts.edu.au/projects/provisioner",
  "@type": "Organization",
  "description": "The University of Technology Sydney Provisioner project is
  "funder": [
    {
      "@id": "https://ror.org/03f0f6041"
    },
    {
      "@id": "https://ands.org.au"
    }
  ],
  "identifier": "https://ereseach.uts.edu.au/projects/provisioner",
  "name": "Provisioner"
},
{
  "@id": "https://ror.org/03f0f6041",
  "@type": "Organisation",
  "identifier": "https://ror.org/03f0f6041",
  "name": "University of Technology Sydney"
},
{
  "@id": "https://ands.org.au",

```

```

"@type": "Organization",
"description": "The core purpose of the Australian National Data Service (A
"identifier": "https://ands.org.au",
"name": "Australian National Data Service"
},

```

See an example of a Dataset which has three listed funders in a chained relationship, for example this Organization is linked as a funder and in turn links to another funder.

## Licensing, Access control and copyright

If a *Data Entity* has a [license](#) that is different from the license on the *Root Data Entity*, the entity SHOULD have a [license](#) property referencing a *Context Entity* with a value of [CreativeWork](#) that describes the license. The ID of the license SHOULD be its URL (e.g. a Creative Commons License URL) and a summary of the license included using the [description](#) property. If this is not possible a URL MAY be used as the value.

This *Data Entity* has a copyright holder which is different from its author. There is a reference to an [Organization](#) describing the copyright holder and a [sameAs](#) relation to a web page.

```

{
  "@id": "SciDataCon Presentations/AAA_Pilot_Project_Abstract.html",
  "@type": "File",
  "contentSize": "17085",
  "copyrightHolder": {
    "@id": "IDRC"
  },
  "author": {
    "@id": "https://orcid.org/0000-0002-0068-716X"
  },
  "description": "Abstract for the Pilot Project initial findings",
  "encodingFormat": "text/HTML",
  "license": {
    "@id": "https://creativecommons.org/licenses/by/4.0/"
  },
  "sameAs": "https://www.scidatacon.org/2016/sessions/56/paper/265/"
},

{
  "@id": "https://orcid.org/0000-0002-0068-716X",
  "@type": "Person",
  "identifier": "https://orcid.org/0000-0002-0068-716X",
  "name": "Cameron Neylon"
},

{
  "@id": "IDRC",
  "@type": "Organization",
  "description": "Canadian Frown Corporation and funder of development research",
  "identifier": "IDRC",

```

```
"name": "International Development Research Center",
},
```

See the example of an [audio recording with a copyrightHolder property](#) (which is different from the author property) and which is linked to a [license](#).

## Equipment

To specify which equipment was used to create or update a *Data Entity*, the *RO-Crate* JSON-LD SHOULD have a *Context Entity* for each item of equipment which SHOULD be of [@type IndividualProduct](#). The entity SHOULD have a serial number, manufacturer that identifies it as completely as possible. In this case the equipment is a bespoke machine. The equipment SHOULD be described on a web page, and the address of the description SHOULD be used as its [@id](#).

```
{
  "@id": "https://confluence.csiro.au/display/ASL/Hovermap",
  "@type": "IndividualProduct",
  "description": "The CSIRO bentwing is an unmanned aerial vehicle (UAV, comm",
  "identifier": "https://confluence.csiro.au/display/ASL/Hovermap",
  "name": "bentwing"
}
```

Uses [CreateAction](#) and [UpdateAction](#) class to model the contributions of *Context Entities* of type [Person](#) or [Organization](#) in the creation of files.

In this example the [CreateAction](#) has a human [agent](#), the object is a [Place](#) (a cave) and the Hovermap drone is the [instrument](#) used in the file creation event.

```
{
  "@id": "DataCapture_wcc02",
  "@type": "CreateAction",
  "agent": {
    "@id": "http://orcid.org/0000-0002-1672-552X"
  },
  "identifier": "DataCapture_wcc02",
  "instrument": {
    "@id": "https://confluence.csiro.au/display/ASL/Hovermap"
  },
  "object": {
    "@id": "victoria_arch"
  },
  "result": [
    {
      "@id": "wcc02_arch.laz"
    },
    {
      "@id": "wcc02_arch_traj.txt"
    }
  ]
},
```

```

{
  "@id": "victoria_arch",
  "@type": "Place",
  "address": "Wombeyan Caves, NSW 2580",
  "description": "This is the GDA94 datum, which is for most situations i
  "geo": {
    "@id": "08563107-deb2-4d3b-a25f-83a09b5b61d4"
  },
  "identifier": "victoria_arch",
  "name": "Victoria Arch"
},

```

See an example of a data-capture [CreateAction](#) with an [Object](#) and two files as results.

## Software

To specify which software was used to create or update a file the software application SHOULD be represented with an entity of type [SoftwareApplication](#), with a [version](#) property. For example:

```

{
  "@id": "https://www.imagemagick.org/",
  "@type": [
    "SoftwareApplication"
  ],
  "url": "https://www.imagemagick.org/",
  "identifier": "https://www.imagemagick.org/",
  "name": "ImageMagick",
  "version": " ImageMagick 7.0.8-2 Q16 x86_64 2018-06-19"
}

```

The software SHOULD be associated with the [File](#) it created using a [CreateAction](#) with the [File](#) referenced by a [result](#) property. Any input files SHOULD be referenced by the [object](#) property.

In the below example, an image with the @id of pics/2017-06-11\_12.56.14.jpg was transformed into a new image pics/sepia\_fence.jpg using the ImageMagick software application. Actions MAY have human-readable names, which MAY be machine generated for use at scale.

```

{
  "@id": "Photo_Capture_1",
  "@type": "CreateAction",
  "agent": {
    "@id": "https://orcid.org/0000-0002-3545-944X"
  },
  "description": "Photo snapped on a photo walk on a misty day",
  "identifier": "Photo1",
  "endTime": "2017:06:11T12:56:14+10:00",
  "instrument": [
    {
      "@id": "EPL1"
    }
  ]
}

```

```

    },
    {
      "@id": "Panny20mm"
    }
  ],
  "result": {
    "@id": "pics/2017-06-11 12.56.14.jpg"
  }
},
{
  "@id": "SepiaConversion_1",
  "@type": "CreateAction",
  "name": "Convert dog image to sepia",
  "description": "convert -sepia-tone 80% test_data/sample/pics/2017-06-11 12.56.14.jpg",
  "identifier": "SepiaConversion",
  "endTime": "2018:09:19T17:01:07+10:00",
  "instrument": {
    "@id": "https://www.imagemagick.org/"
  },
  "object": {
    "@id": "pics/2017-06-11 12.56.14.jpg"
  },
  "result": {
    "@id": "pics/sepia_fence.jpg"
  }
}
},

```

See an [example](#) of a [CreateAction](#), with a result which is a [File](#) and an [object](#), which is a place.

## Workflows and scripts

Scientific workflows and scripts that can be or were used to analyze or generate files contained in an the RO-Crate MAY be embedded in an RO-Crate. Workflows and scripts are described using the type `SoftwareSourceCode` instead of `SoftwareApplication` above.

The distinction is fluid, and comes down to availability and understandability. For instance, office spreadsheet applications are generally available and do not need further explanation; while a Python script that is customized for a particular data analysis might be important to understand further and should be included as source code in the RO-Crate dataset:

```

{
  "@id": "workflow/retropath.knime",
  "@type": "SoftwareSourceCode",
  "additionalType": {"@id": "Workflow"},
  "name": "RetroPath Knime workflow",
  "description": "KNIME implementation of RetroPath2.0 workflow",
  "author": {"@id": "#thomas"},
  "programmingLanguage": {"@id": "#knime"},
  "license": "https://spdx.org/licenses/BSD-2-Clause.html",
  "potentialAction": {
    "@type": "ActivateAction",

```

```
      "instrument": {"@id": "#knime"}
    }
  }
```

Workflows and scripts are saved on disk using a language and generally need a runtime, in RO-Crate this SHOULD be indicated using a liberal interpretation of `programmingLanguage` and `potentialAction`. Note that the language and its runtime might differ, e.g. there are multiple potential runtimes of a JavaScript, although in the base-case they are the same, and as a shortcut can be described in one go as both a `ComputerLanguage` and `SoftwareApplication`:

```
{
  "@id": "#knime",
  "@type": [
    "ComputerLanguage",
    "SoftwareApplication"
  ],
  "name": "KNIME Analytics Platform",
  "alternateName": "KNIME",
  "url": {"@id": "https://www.knime.com/knime-software/knime-analyt"},
  "version": "3.6"
}
```

A data entity representing a runtime application or language MUST have a [name](#), [url](#) and [version](#), which should indicate a known version the workflow/script was developed or tested with. [alternateName](#) MAY be provided if there is a shorter colloquial name, for instance “R” instead of “The R Project for Statistical Computing”.

To indicate that a `SoftwareSourceCode` is in the form of a script (e.g. sequential batch/shell script), use:

```
"additionalType": {"@id": "WorkflowScript"}
```

If the `SoftwareSourceCode` is in the form of a workflow (e.g. a pipeline of steps with data flow), use:

```
"additionalType": {"@id": "Workflow"}
```

### Workflow diagram/sketch

It can be beneficial to show a diagram or sketch to explain the script/workflow. This may have been generated from the workflow management system or drawn manually as a diagram. It should be included as an `ImageObject` which is about the `SoftwareSourceCode`:

```
{
  "@id": "workflow/workflow.svg",
  "@type": "ImageObject",
  "additionalType": "WorkflowSketch",
  "encodingFormat": "image/svg+xml",
  "description": "Diagram of RetroPath2.0 workflow",
}
```



```
    "about": {"@id": "workflow/workflow.knime"}
  }
```

The image file format SHOULD be indicated with [encodingFormat](#) via Mime - eg "image/svg+xml".

Additionally a PRONOM identifier MAY be used, via a reference to a *ContextualEntity* {"@id": "https://www.nationalarchives.gov.uk/PRONOM/fmt/92"}

```
{
  "@id": "workflow/workflow.svg",
  "@type": "ImageObject",
  "additionalType": "WorkflowSketch",
  "encodingFormat": ["image/svg+xml", {"@id": "https://www.nationa
  "description": "Diagram of RetroPath2.0 workflow",
  "about": {"@id": "workflow/workflow.knime"}
}
{
  "@id": "https://www.nationalarchives.gov.uk/PRONOM/fmt/92", "@type": "URL"
  "name": "Portable Network Graphics",
  "Description": "Portable Network Graphics (PNG) ... "
}
```

A *Sketch* may still be provided even if there is no programmatic *SoftwareSourceCode* that can be executed (e.g. because the workflow was done by hand). In this case the sketch should have an *about* referring to the *RO-Crate dataset* as a whole:

```
"about": {"@id": "./"}
```

## Representing Provenance: changes to an object

To record an action which changes the *DataSet*'s metadata, or changes its state in a publication or other workflow, a [CreateAction](#) or [UpdateAction](#) SHOULD be associated with a *Data Entity*.

A curation Action MUST have at least one [object](#) which associates it with either the *DataSet* or one of its components.

An Action which creates new *Data entities* - for example, the creation of a new metadata file - SHOULD have these as [results](#).

An Action SHOULD have a [name](#) and MAY have a [description](#).

An Action SHOULD have an [endTime](#), which MUST be in ISO 8601 date format and SHOULD be specified to at least the precision of a day. An Action MAY have a [startTime](#) meeting the same specifications.

An Action SHOULD have a human [agent](#) who was responsible for authorizing the action, and MAY have an [instrument](#) which associates the action with a particular piece of software (for example, the content management system or data catalogue through which an update was approved) which SHOULD be of [@type SoftwareApplication](#).

An Action's status MAY be recorded in an [actionStatus](#) property. The status must be one of the values enumerated by [ActionStatusType](#): [ActiveActionStatus](#), [CompletedActionStatus](#), [FailedActionStatus](#) or [PotentialActionStatus](#).

An Action which has failed MAY record any error information in an [error](#) property.

[UpdateAction](#) SHOULD only be used for actions which affect the DataSet as a whole, such as movement through a workflow.

To record curation actions which modify a [File](#) within a DataSet - for example, by correcting or enhancing metadata - the old version of the [File](#) SHOULD be retained, and a [CreateAction](#) added which has the original version as its [object](#) and the new version as its [result](#).

```
{
  "@id": "history-01",
  "@type": "CreateAction",
  "object": { "@id": "https://doi.org/10.5281/zenodo.1009240" },
  "name": "R0-Crate created",
  "endTime": "2018-08-31",
  "agent": { "@id": "https://orcid.org/0000-0001-5152-5307" },
  "instrument": { "@id": "https://stash.research.uts.edu.au" },
  "actionStatus": "CompletedActionStatus"
},

{
  "@id": "history-02",
  "@type": "UpdateAction",
  "object": { "@id": "https://doi.org/10.5281/zenodo.1009240" },
  "name": "R0-Crate published",
  "endTime": "2018-09-10",
  "agent": { "@id": "https://orcid.org/0000-0001-5152-5307" },
  "instrument": { "@id": "https://stash.research.uts.edu.au" },
  "actionStatus": "CompletedActionStatus"
},

{
  "@id": "history-03",
  "@type": "CreateAction",
  "object": { "@id": "./metadata.xml.v0.1" },
  "result": { "@id": "./metadata.xml" },
  "name": "metadata update",
  "endTime": "2018-09-12",
  "agent": { "@id": "https://orcid.org/0000-0001-5152-5307" },
  "instrument": { "@id": "https://stash.research.uts.edu.au" },
  "actionStatus": "CompletedActionStatus"
},

{
  "@id": "history-04",
  "@type": "UpdateAction",
  "object": { "@id": "https://doi.org/10.5281/zenodo.1009240" },
  "name": "R0-Crate published",
```

```

    "endTime": "2018-09-13",
    "agent": { "@id": "https://orcid.org/0000-0001-5152-5307" },
    "instrument": { "@id": "https://stash.research.uts.edu.au" },
    "actionStatus": "FailedActionStatus",
    "error": "Record is already published"
  },

  {
    "@id": "https://stash.research.uts.edu.au",
    "@type": "IndividualProduct",
    "name": "Stash",
    "description": "UTS Research Data Catalogue",
    "identifier": "https://stash.research.uts.edu.au"
  }
}

```

### Extra metadata such as Exif

Schema.org has a generic extension mechanism for encoding adding arbitrary properties and values which are not available as Schema.org properties. An example of this is the Schema.org [recommended way \(see example 2\)](#) of including [Exif](#) technical image metadata.

To include EXIF, or other data which can be encoded as property/value pairs, add an array of references to *Anonymous Entities* which encode each property. This example shows one property of several hundred.

```

{
  "@id": "pics/2017-06-11 12.56.14.jpg",
  "@type": "ImageObject",
  "contentSize": "5114778",
  "author": {
    "@id": "https://orcid.org/0000-0002-3545-944X"
  },
  "description": "Depicts a fence at a disused motor racing venue with the",
  "encodingFormat": "Exchangeable Image File Format (Compressed)",
  "exifData": [
    {
      "@id": "_:b0"
    },
    {
      "@id": "_:b312"
    }
  ]
}

{
  "@id": "_:b312",
  "@type": "PropertyValue",
  "name": "InternalSerialNumber",
  "value": "4102011002108002"
},

```

See an example of Exif data displayed in HTML.

## Places

To associate a *Data Entity* with a *Context Entity* representing a *geographical location or region* the entity SHOULD have a property of `contentLocation` with a value of type `Place`.

This example shows how to define a place, using a `geonames` ID:

```
{
  "@id": "https://www.geonames.org/8152662/catalina-park.html",
  "@type": "Place",
  "description": "Catalina Park is a disused motor racing venue, located at K
  "geo": {
    "@id": "b4168a98-8534-4c6d-a568-64a55157b656"
  },
  "identifier": "https://www.geonames.org/8152662/catalina-park.html",
  "name": "Catalina Park"
},
```

The place has a `geo` property, referencing an *Context Entity* of @type `GeoCoordinates`:

```
{
  "@id": "b4168a98-8534-4c6d-a568-64a55157b656",
  "@type": "GeoCoordinates",
  "latitude": "-33.7152",
  "longitude": "150.30119",
  "name": "Latitude: -33.7152 Longitude: 150.30119"
},
```

The `GeoCoordinates` item SHOULD have a human readable `name`, which is used in generating the `ro-crate-preview.html` file.

And the place is referenced from the `contentLocation` property of the dataset.

```
{
  "@id": "https://doi.org/10.5281/zenodo.1009240",
  "@type": "Dataset",
  "outputOf": "R0-Crate",
  "contact": {
    "@id": "https://orcid.org/0000-0002-3545-944X"
  },
  "contentLocation": {
    "@id": "https://www.geonames.org/8152662/catalina-park.html"
  },
},
```

`Place` MAY use any of the [resources available in Schema.org](#) to describe places. Future profiles of RO-Crate may mandate the use of a subset of these. Any directory or file or *Context Entity* may be geo-located. For example this file:

```
{
  "@id": "pics/19093074_10155469333581584_5707039334816454031_o.jpg",
  "@type": "File",
  "contentLocation": {
    "@id": "https://www.geonames.org/8152662/catalina-park.html"
  },
  "contentSize": "132765",
  "author": {
    "@id": "https://orcid.org/0000-0002-3545-944X"
  },
}
```

See an example of a place in [HTML](#).

## Time

To describe the time period which a RO-Crate Data Entity is *about*, use [temporalCoverage](#).

## Subjects & keywords

Subject properties (equivalent to a Dublin Core Subject) on RO-Crate MUST use the [about](#) property.

Keyword properties MUST use [keyword](#). Note that by schema.org convention keywords are given as a single JSON string, with individual keywords separated by commas.

```
{
  "keyword": "Gibraltar, Spain, British Overseas Territory, city, map",
  "about": { "@id": "http://dbpedia.org/resource/Gibraltar" },
}
```

## Digital Library and Repository content

To describe an export from a Digital Library or repository system, RO-Crate uses the *Portland Common Data Model (PCDM)*. A *Context Entity* from a repository, representing an abstract entity such as a person, or a work, or a place SHOULD have a@type of [RepositoryObject](#), in addition to any other types. Objects MAY be grouped together in [RepositoryCollections](#) with [hasMember](#) pointing to the [RepositoryObject](#). The keys [RepositoryObject](#) and [RepositoryCollection](#) were chosen to avoid collision between the terms [Collection](#) and [Object](#) with other vocabularies.

NOTE: [PCDM](#) specifies that Files should have only technical metadata, not descriptive metadata, which is *not* a restriction in RO-Crate. If the RO-Crate is to be imported into a strict [PCDM](#) repository, modeling of object/file relationships will be necessary.

For example, this data is exported from an [Omeka](#) repository:

```
{
  "@id": "https://omeka.uws.edu.au/farmstofreeways/api/collections/6",
  "@type": [
    "RepositoryCollection"
  ],
  "title": [
    "Project Materials"
  ],
}
```

```

"description": [
  "Materials associated with the project, including fliers seeking partic
],
"publisher": [
  "University of Western Sydney"
],
"rights": [
  "Copyright University of Western Sydney 2015"
],
"hasMember": [
  {
    "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/166"
  },
  {
    "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/167"
  },
  {
    "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/168"
  },
  {
    "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/169"
  }
]
},
{
"@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/166",
"@type": [
  "RepositoryObject",
  "Text"
],
"title": [
  "Western Sydney Women's Oral History Project: Flier (illustrated)"
],
"description": [
  "Flier (illustrated) seeking participants for the project."
],
"publisher": [
  "University of Western Sydney"
],
"rights": [
  "Copyright University of Western Sydney 2015"
],
"originalFormat": [
  "Paper"
],
"identifier": [
  "FTF_flier_illust"
],
"rightsHolder": [
  "Western Sydney University"
],
"license": [

```

```

    "Content in the Western Sydney Women's Oral History Project: From farms
  ],
  "hasFile": [
    {
      "@id": "./content/166/original_eece70f73bf8979c0bcfb97065948531.pdf"
    },
    ...
  ]
},
{
  "@type": "File",
  "@id": "./content/166/original_eece70f73bf8979c0bcfb97065948531.pdf"
},

```

See an example of a repository exported to a RO-Crate.

## Thumbnails

A [File](#) or any other item MAY have a [thumbnail](#) property which references another file.

For example, the below [RepositoryObject](#) is related to four files which are all versions of the same image (via [hasFile](#)) one of which is a thumbnail. The thumbnail MUST be included in the RO-Crate.

If [thumbnails](#) are incidental to the data set, they need not be referenced by [hasPart](#) or [hasFile](#) relationships. but must be in the BagIt manifest if in a *Bagged RO-Crate*.

```

{
  "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/383",
  "@type": [
    "RepositoryObject",
    "Image"
  ],
  "identifier": [
    "ftf_photo_stapleton1"
  ],
  "interviewee": [
    {
      "@id": "https://omeka.uws.edu.au/farmstofreeways/api/items/595",
    }
  ],
  "description": [
    "Photo of Eugenie Stapleton inside her home"
  ],
  "license": [
    "Content in the Western Sydney Women's Oral History Project: From farms t
  ],
  "publisher": [
    "University of Western Sydney"
  ],
  "hasFile": [
    {

```

```
"@id": "files/383/original_c0f1189ec13ca936e8f556161663d4ba.jpg"
},
{
  "@id": "files/383/fullsize_c0f1189ec13ca936e8f556161663d4ba.jpg"
},
{
  "@id": "files/383/thumbnail_c0f1189ec13ca936e8f556161663d4ba.jpg"
},
{
  "@id": "files/383/square_thumbnail_c0f1189ec13ca936e8f556161663d4ba.jpg"
}
],
"thumbnail": [
  {
    "@id": "files/383/thumbnail_c0f1189ec13ca936e8f556161663d4ba.jpg"
  }
],
"name": [
  "Photo of Eugenie Stapleton 1"
],
"relatedLink": [
  "<a href=\"https://omeka.uws.edu.au/farmstofreeways/items/show/512\">Audio"
],
"copyrightHolder": [
  {
    "@id": "https://westernsydney.edu.au",
  }
],
"copyright": [
  "Copyright University of Western Sydney 2015"
]
},
{
  "@type": "File",
  "@id": "files/384/original_2ebbe681aa6ec138776343974ce8a3dd.jpg"
},
{
  "@type": "File",
  "@id": "files/384/fullsize_2ebbe681aa6ec138776343974ce8a3dd.jpg"
},
{
  "@type": "File",
  "@id": "files/384/thumbnail_2ebbe681aa6ec138776343974ce8a3dd.jpg"
},
{
  "@type": "File",
  "@id": "files/384/square_thumbnail_2ebbe681aa6ec138776343974ce8a3dd.jpg"
},
},
```



[See an example of a thumbnail.](#)

## Extending RO-Crate

To extend RO-Crate implementers SHOULD try to use valid schema.org properties and classes and MAY use terms from other vocabularies and ontologies when this is not possible.

The terms (properties and classes) used SHOULD be added as keys to the @context in the *RO-Crate JSON-LD*. URIs in the @context SHOULD resolve to a useful human readable page. Where this is not possible - for example if the URI resolves to an RDF ontology file a human-readable URI SHOULD be provided using SameAs.

For example. This URI from the [BIBO](#) ontology resolves to an ontology file, which is not useful for humans, but [this page](#) is human-readable:

```
{
  "@context": [ "https://w3id.org/ro/crate/0.2/context"
    { "interviewee": "http://purl.org/ontology/bibo/interviewee" },
  ],
  "@graph": [
    {
      "@id": "http://purl.org/ontology/bibo/interviewee",
      "sameAs": "http://neologism.ecs.soton.ac.uk/bibo.html#interviewee",
      "@type": "URL"
    }
  ]
}
```

When generating the *RO-Crate Website* from *RO-Crate JSON-LD*, the code MUST use a sameAs URI (if present) as a target for an explanatory link for the term instead of the Linked Data URI supplied in the @context.

[See an example of the key "interviewee" resolving to a human-readable page.](#) and download the [RO-Crate JSON-LD](#)

Where there is no RDF ontology available, then implementors SHOULD attempt to provide context by creating stable web-accessible URIs to document properties and classes, for example, by linking to page describing an XML element or attribute from an XML schema, pending the publication of a formal ontology.

---

This project is maintained by [ResearchObject](#)

Hosted on GitHub Pages — Theme by [orderedlist](#)