researchobject.org Specification

Research Object Bundle 1.0

researchobject.org Specification 05 November 2014

This version:
https://w3id.org/bundle/2014-11-05/
Previous version:
http://purl.org/wf4ever/ro-bundle/2013-05-21/
Latest editor's draft:
https://w3id.org/bundle/draft/
Latest Recommendation:
https://w3id.org/bundle/
Editors:
Stian Soiland-Reyes, University of Manchester
Matthew Gamble, University of Manchester
Authors:
Stian Soiland-Reyes, University of Manchester
Matthew Gamble, University of Manchester
Robert Haines, University of Manchester

DOI 10.5281/zenodo.12586

Abstract

This specification defines a file format for storage and distribution of Research Objects as a ZIP archive; called a **Research Object Bundle** (RO Bundle). RO Bundles allow capturing a Research Object to a single file or byte-stream by including its manifest, annotations and some or all of its aggregated resources for the purposes of exporting, archiving, publishing and transferring research objects.

Status of This Document

This document is a specification published by researchobject.org and and does not represent the support or consensus of any standards organisation.

Questions, feedback and comments are kindly requested, either as GitHub issues, or as pull requests of the current draft.

This document uses the terms [URI] and [IRI] interchangably. As long as the implementing technology supports it, all uses of URIs can be interchanged with IRIs (e.g. URIs containing unescaped Unicode characters).

Table of Contents

1. Introduction

This section is non-normative.

The Wf4Ever Research Object model [RO] defines a model for aggregating the resources that contribute to a scientific work, including domain-specific annotations and provenance traces. The unit that collects these resources is called a *Research Object* (RO) and is described in an RDF-based manifest according to the Research Object OWL ontologies. The RO model has been formed in particular for the purpose of preservation of scientific workflows, but is applicable also in a general sense for capturing digital resources that are related to eachother, and which together form a trackable whole. The Research Object primer [ROPrimer] provides further details and examples of using the RO model.

The specification for the RO model does not mandate any particular form for the representation of Research Objects. The Wf4Ever RO API [ROAPI] defines how research objects can be accessed and maintained on the web through a RESTful web service exposing RDF/XML and Turtle representations. Practical use of the RO model has however shown that it is also benefitial to represent a research object as a single ZIP archive or as file system folders for the purposes of downloading, editing and archiving a research object.

For instance a scientific workflow system can export a workflow run by saving the workflow definition, runtime provenance trace and generated results to a set of files. A research object that represents the workflow run can aggregate and relate these resources. However, at the time of running the workflow (e.g. on a desktop computer) it is often not known where or if the user would choose to publish the RO; thus the direct use of a web service or minting public URIs is problematic in this situation.

A **Research Object Bundle**, as specified by this document, provides a way to collect the resources that are aggregated in a research object, represented as files in a ZIP archive, in addition to their metadata and annotations. The ZIP archive thus becomes a single representation of a research object and which can be exported, archived, published and transferred like a regular file or resource.

2. Container

A Research Object Bundle is a structured [ZIP] archive, specializing the Adobe Universal Container Format [UCF]. UCF is based on the EPUB [OCF] format, but generalized to be any kind of container. The following section gives an informal introduction to the UCF format. For the complete, normative details, see the [UCF] specification.

2.1 Universal Container Format (UCF)

This section is non-normative.

A UCF container is based on the ZIP compression file format [ZIP], enforcing additional restrictions. The most important restrictions are:

- Reserved filenames in the root directory: `mimetype` and `META-INF`
- Filenames must be encoded in UTF-8
- Compression must be Uncompressed or Flate
- MAY use Zip64 extensions, but SHOULD only do so when required
- The first file MUST be the uncompressed `mimetype` and without any extra attributes

UCF says about `mimetype`:

> *The first file in the Zip container* MUST *be a file with the ASCII name of* `mimetype`*, which holds the MIME type for the Zip container (~~application/epub+zip~~ as an ASCII string; no padding, white-space, or case change).*

The actual media type to include in `mimetype` depends on the specific container type (the above quote uses ePub as an example). See section 2.2 RO bundle container.

> ***Best Practice 1: Use zip -0 -X***
>
> To add the `mimetype` file correctly on a UNIX/Linux installation with InfoZip, use `echo -n` and `zip -0 -X`. Below is an example which adds `mimetype` correctly as the first, uncompressed file, then the remaining files (excluding `mimetype`) with the default compression:

EXAMPLE 1

```
stain@ahtissuntu:~/test$ echo -n application/vnd.wf4ever.robundle+zip > mimetype

stain@ahtissuntu:~/test$ zip -0 -X ../example.robundle mimetype
  adding: mimetype (stored 0%)

stain@ahtissuntu:~/test$ zip -X -r ../example.robundle . -x mimetype
  adding: META-INF/ (stored 0%)
  adding: META-INF/container.xml (stored 0%)
  adding: .ro/ (stored 0%)
  adding: .ro/manifest.json (stored 0%)
  adding: helloworld.txt (stored 0%)
```

2.1.1 Rootfile

A *root file* is the entry-point for a UCF container, playing a similar role to `index.html` on web servers.

UCF says about `META-INF/container.xml` and rootfiles:

> *A UCF Container* MAY *include a file named* `container.xml` *in the* `META-INF` *directory at the root level of the container file system. If present, the* `container.xml` *file* MAY *identify the MIME type of, and path to, the root file for the container and any* OPTIONAL *alternative renditions included in the container.*

An example of `META-INF/container.xml` which defines the *rootfile* as `.ro/manifest.json`:

EXAMPLE 2

```
<?xml version="1.0"?>
<container version="1.0"
    xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
    <rootfiles>
        <rootfile full-path=".ro/manifest.json" media-type="application/ld+json" />
    </rootfiles>
</container>
```

2.2 RO bundle container

The RO Bundle container is a specialization of a [UCF] container, with the following additions:

- Additional reserved filename in the root directory: `.ro` MUST be present and MUST be a directory.
- The `mimetype` SHOULD be `application/vnd.wf4ever.robundle+zip` (see below)
- The `META-INF/container.xml`, if present, SHOULD contain a rootfile entry equivalent to:
  `<rootfile full-path=".ro/manifest.json" media-type="application/ld+json" />`
  If the container file is missing, the above rootfile entry SHOULD be assumed.
- The file `.ro/manifest.json` MUST be present, and MUST describe the RO according to section 3. Manifest.

Applications who specialize RO Bundles MAY specify a different `mimetype`, for instance because the bundle is used to distribute application-specific data. It is RECOMMENDED for such extensions that their media type end with +zip according to [RFC6839] unless it is not considered meaningful for a user to treat such bundles as a general ZIP archive.

2.2.1 Resource media type

If an application requires a media-type for a resource, for instance because it is exposing the RO bundle over HTTP, it SHOULD resolve the media type of the resource according to this section.

In order of preference:

1. A resource which is a root file is assumed to have the media type given by the `mimetype` of the corresponding (or implied) `<rootfile>` entry.
2. If a resource is an external reference (e.g. referenced with an absolute `http://` URI), then its media type is given by the HTTP `Content-Type`, which may involve content negotiation.
3. If the resource is a aggregated in the manifest, applications SHOULD use the `mediatype` (`dc:format` in RDF manifests), if present.
4. Failing the above, the media type of a resource MAY be resolved according to the following table by case-insensitive matching of its extension (suffix):

   Extension Media type

   `.txt` `text/plain; charset="utf-8"`

`.ttl`	`text/turtle; charset="utf-8"`
`.rdf`	`application/rdf+xml`
`.json`	`application/json`
`.jsonld`	`application/ld+json`
`.xml`	`application/xml`

5. In the absence of a resolved media type, the media type `application/octet-stream` MAY be assumed.

2.2.2 META-INF/manifest.xml

To avoid confusion with the somewhat overlapping RO manifest it is NOT RECOMMENDED to include the ODF manifest (`META-INF/manifest.xml`) in RO Bundles or to use the [ODF] manifest for resolving media types.

3. Manifest

The research object MUST be described in the file `.ro/manifest.json` as specified below. Alternative manifests MAY also be present.

3.1 .ro/manifest.json

The file `.ro/manifest.json`, MUST contain the [ORE] manifest for the research object according to this section. The file MUST be in JSON format [RFC4627], and SHOULD be valid [JSON-LD].

Identifiers used below are either:

1. *Meta-resources*, path relative to `/.ro/` directory, which MUST NOT contain the `:` character. For instance `manifest.json` or `annotations/ann2`. Depending on how meta-resources are used, the ZIP might or might not include a corresponding entry for the given path.
2. *Bundled resources*, the path SHOULD start with `/` to indicate the root of the bundle, for instance `/hello.txt` or `/folder2/`. Folders SHOULD have a path terminating with `/`. The resource identified by the path SHOULD be included as a corresponding file or directory in the ZIP file.
3. *Absolute URIs* (contains `:`), external to the bundle. For instance `http://www.example.com/external`

Identifiers with special characters (e.g. space) MUST be URI escaped, while international characters (e.g. Unicode) MAY be escaped or expressed as an [IRI].

3.1.1 JSON structure

The structure of the JSON manifest is given by an JSON Object with the members:

`@context`
: JSON-LD context. SHOULD be present, in which case it MUST be valid according to the JSON-LD @context keyword. The RO bundle context SHOULD be a list, and SHOULD include the value `"https://w3id.org/bundle/context"`. This value SHOULD be the last item of the list.

`id`
: RO identifier. SHOULD be present, in which case it SHOULD have the fixed value `/` indicating the relative top-level folder as the identifier. (See section 4. Identifiers.) If the RO bundle was downloaded from a web resource, then the URI it was retrieved from SHOULD be provided with `retrievedFrom`.

`manifest`
: ORE manifests describing this RO, relative to the `.ro/` directory. SHOULD be literal `manifest.json`, but MAY be a list, in which case the list MUST contain `manifest.json`

`history`
: Provenance trace of the life of this RO, relative to the `.ro/` directory. This property MAY be present, in which case it SHOULD be `evolution.ttl`, indicating that the bundle file `/.ro/evolution.ttl` contains the provenance trace. This value MAY be an absolute URI. The property MAY give a list if several provenance traces are known, in which case the list SHOULD include `evolution.ttl`.

  The file `/.ro/evolution.ttl`, if present, SHOULD include a provenance trace of this research object according to the roevo ontology.

  The higher level provenance of the research object (e.g. it's creator and creation date) SHOULD be provided as additional members, even if the `history` member is present.

`aggregates`
: This property SHOULD be present, in which case it MUST be a list of *all* the resources aggregated by this RO. The values in a list MUST be objects, which MUST be uniquely identified by `uri`. The members are:

  `uri`
  : A path within the bundle, or an absolute URI. Bundle paths SHOULD be prefixed with `/` unless they are relative to the `/.ro/` folder. (See prefixes.) Special characters such as space MUST be URI escaped, while international characters (e.g. Unicode) MAY be escaped or expressed as an [IRI].

  `mediatype`
  : The IANA media type of the (typically identified by `file`) resource. This SHOULD be specified for resource identified by a bundle path unless its media type is correctly identified according to section 2.2.1 Resource media type.

  `conformsTo`
  : The URI of a specification, standard, schema, vocabulary or similar that the resource conforms to. This member MAY be provided, in which case the value SHOULD be an absolute URI identifying a versioned, retrievable specification that the resource conforms to. Example: `http://www.w3.org/TR/SVG11/`

  `bundledAs`
  : An ORE proxy [ORE], providing details of how this resource has been bundled. This object SHOULD be present for aggregated resources aggregated with an absolute URI, and MAY be present for other aggregated resources. Its members are:

    `uri`
    : The identifier for the ORE Proxy as an URI. This property MUST be provided if the `bundledAs` object exists. This identifier should be used if referring to "resource X as aggregated in research object Y" within annotations and in external documents. The proxy identifier SHOULD consist of the prefix `urn:uuid:` and a lowercased UUID string [RFC4122], for example: `urn:uuid:d4f09040-272e-467f-9250-59593bd4ac8f`

    `folder`
    : A folder in the bundle this resource belongs to. This member MUST be present if `filename` is given, and MAY be present alone. The path SHOULD be prefixed with `/` and SHOULD end with `/`, for instance `/folder2/` or `/`. The folder SHOULD be a directory in the zip archive.

    `filename`
    : The filename the resource is given within the specified folder. This member SHOULD be present if `folder` is also given. The filename should not contain the characters `/`, `:` or `\`, but MAY contain spaces and international characters.

    Additional members detailing the provenance of the proxy (i.e. describing who aggregated the resource) MAY be included (see below). Other metadata about a proxy (e.g. comments about why a resource was included), if present, SHOULD be added as an annotation (see below) using the proxy identifier as `about`.

  Additional members detailing the provenance of every aggregated resource SHOULD be included (see below). Other metadata about a resource (e.g. a title or description), if present, SHOULD be added as an annotation (see below) using the resource `uri` as `about`.

  The order of the values in the `aggregates` list is insignificant, however the list MUST NOT contain duplicate entries. An entry is considered duplicate by comparing the `uri` value resolved as an unescaped and absolute IRI.

`annotations`
: Annotations about this research object and its resources. This member MAY be present, in which case it member MUST be a list. An annotation [OA] provides additional metadata or descriptions which are somewhat about or related to the research object or some of its aggregated resources.

  An annotation is specified as an object, which have the following members:

`uri`
: The identifier for this annotation. The identifier SHOULD be present, and SHOULD consist of the prefix `urn:uuid:` and a lowercased UUID string [RFC4122]. For example: `urn:uuid:1a876f9e-4ffe-4c99-a05d-cd9d0cbd4cbb`

`about`
: The identifier for the annotated resource, MUST be present. This is considered the target of the annotation, the resource the annotation content is "somewhat about". The "about" identifier SHOULD be one of these types:
  - The research object itself, which MUST match the value of its `id`, e.g. `"/"`.
  - An aggregated resource, matching its `uri` under `aggregates`.
  - A proxy for an aggregated resource, which MUST match the `uri` of `bundledAs` on an aggregated resource.
  - Another annotation, which MUST match the `uri` as listed under `annotations`.
  - An absolute URI that is not aggregated by this research object. If the `about` resource is not aggregated, the corresponding `content` member SHOULD be an aggregated resource.
  - A JSON list, containing any of the above. This indicates that the annotation is about each of the listed resources, for instance because the annotation content is describing their relationship.

`content`
: The identifier for a resource that contains the body of the annotation, SHOULD be present. The content identifier SHOULD be one of these types:
  - A (non-aggregated) meta-resource (typically an RDF graph), starting with `annotations/`, which MUST exist in the `/.ro/annotations/` directory.
  - An aggregated resource, matching its `uri` under `aggregates`.
  - An absolute URI, which may or may not be aggregated by the RO. If the content is not aggregated, the corresponding `about` MUST NOT be an absolute URI that is not aggregated by the research object.

Additional properties describing the annotation using the oa: namespace MAY be added according to section 3.3 Custom JSON-LD.

The members `about` or `content` SHOULD identify at least one resource that is otherwise part of the research object (e.g. the research object itself, an aggregated resource, a proxy or another annotation). Annotations are considered implicitly aggregated by a research object, and SHOULD NOT be listed under `aggregates` of the same research object.

`@graph`
: A list of additional [JSON-LD] statements according to section 3.3 Custom JSON-LD.

3.1.2 Provenance information

Provenance information (describing creators, dates and sources) SHOULD be provided for these JSON objects in the manifest:

- The RO (top-level JSON object)
- Aggregated resources (values under `aggregates`)

Provenance information MAY also be provided for these JSON objects in the manifest:

- ORE proxies (values under `bundledAs`)
- Annotations (values under `annotations`)

Provenance information is given by the following members:

`createdOn`
: The time the resource representation was created (e.g. when it was saved from an application). SHOULD be present, in which case it MUST be a xsd:dateTime formatted timestamp (ISO 8601), and SHOULD include the time zone.

`createdBy`
: The creator of the resource representation (e.g. who saved it from an application). The creator is an agent, e.g. a person, organization or software. This MAY be different from the agent who conceptually formed the resource (e.g. wrote the document or chose the aggregated resources), which SHOULD be indicated with `authoredBy`. The creator SHOULD be an object with the following members:

  `name`
  : The full name of the agent. The name MUST be present. Examples: `"John Doe"` or `"University of Manchester"`

  `uri`
  : A URI identifying the agent. The URI SHOULD be present, and SHOULD be a valid WebID, for instance `http://example.com/fred#fred`

  `orcid`
  : An ORCID identifier for this agent. For instance, `http://orcid.org/0000-0001-9842-9718`. An ORCID MAY be present if known, and MUST be a URI.

  Additional [FOAF] properties (such as `foaf:homepage`) MAY be added to according to section 3.3 Custom JSON-LD.

`authoredOn`
: The time the resource was conceptually formed. The author time SHOULD be present if different from `createdOn`. The value MUST be a xsd:dateTime formatted timestamp (ISO 8601), and SHOULD include the time zone.

`authoredBy`
: The author of the resource, i.e. the agent(s) that conceptually formed its content. Examples include who authored the text of an aggregated document, or chose the resources that were aggregated in a research object. The author SHOULD be present if different from `createdBy`.

  The author SHOULD be a JSON object with the same members and requirements as for `createdBy`, but MAY be a list to indicate multiple authors.

`retrievedFrom`
: The absolute URI where a resource has been retrieved from. This property SHOULD be included if a bundle resource was downloaded from an external source. This property SHOULD be accompanied with `retrievedOn` and `retrievedBy`.

`retrievedOn`
: The time the resource was retrieved on. If this property is present, then `retrievedFrom` MUST also be present. The value MUST be a xsd:dateTime formatted timestamp (ISO 8601), and SHOULD include the time zone.

`retrievedBy`
: The agent that the resource was retrievedBy, i.e. the person that downloaded the resource. If this property is present, then `retrievedFrom` MUST also be present. The agent SHOULD be a JSON object with the same members and requirements as for `createdBy`.

Additional provenance (curation, contribution, derivation, etc.) MAY be added using the pav: namespace according to section 3.3 Custom JSON-LD or detailed in a separate history provenance trace.

3.1.3 Example manifest.json

An example of a manifest which is valid JSON-LD is included below:

EXAMPLE 3

```
{
    "@context":  [
        "https://w3id.org/bundle/context"
    ],
    "id": "/",
    "manifest":  "manifest.json",
    "createdOn": "2013-03-05T17:29:03Z",
    "createdBy": {
        "uri":     "http://example.com/foaf#alice",
        "orcid":   "http://orcid.org/0000-0002-1825-0097",
        "name":    "Alice W. Land" },
    "history":   "evolution.ttl",
    "aggregates": [
```

```
        { "uri":  "/folder/soup.jpeg" },
        { "uri":  "http://example.com/blog/" },
        { "uri":      "/README.txt",
          "mediatype": "text/plain",
          "createdBy": {
              "uri":     "http://example.com/foaf#bob",
              "name":    "Bob Builder" },
          "createdOn": "2013-02-12T19:37:32.939Z" },
        { "uri":    "http://example.com/comments.txt",
          "bundledAs": {
             "uri":    "urn:uuid:a0cf8616-bee4-4a71-b21e-c60e6499a644",
             "folder":   "/folder/",
             "filename": "external.txt" }
        }
    ],
    "annotations": [
      { "uri":     "urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf",
        "about":   "/folder/soup.jpeg",
        "content": "annotations/soup-properties.ttl" },

      { "about":   "urn:uuid:a0cf8616-bee4-4a71-b21e-c60e6499a644",
        "content": "http://example.com/blog/they-aggregated-our-file" },

      { "about":   [ "/", "urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf" ],
        "content": "annotations/a-meta-annotation-in-this-ro.txt" }
    ]
}
```

3.2 JSON-LD and mapping to RO model

Manifests following the JSON structure defined in section 3.1 .ro/manifest.json with a `"@context": [ "https://w3id.org/bundle/context" ]` is intended to be valid [JSON-LD] without any additional modifications. Mapping `.ro/manifest.json` to the ORE and [RO] models in RDF SHOULD be performed according to the algorithm for conversion from JSON to RDF, as specified in the JSON-LD API [JSON-LD].

In order to generate the RDF implied by the manifest, an absolute base URI SHOULD be assumed according to section 4.2 Absolute URIs for bundle resources with a path element of `/.ro/manifest.json`. This is to ensure that paths starting with `/` don't "climb out" of the bundle root, and so that relative paths like `annotations/soup-properties.ttl` are resolved correctly within the `/.ro/` directory. For example, generating a random UUID for the RO bundle and assuming a base URI of `app://9c13b6c1-d053-4889-b42d-691b4a54338e/.ro/manifest.json` then the bundled resource `/folder/soup.jpeg` will be represented in the RDF graph as `app://9c13b6c1-d053-4889-b42d-691b4a54338e/folder/soup.jpeg`.

The content of https://w3id.org/bundle/context at time of writing is:

```
{
  "@context": {
    "ao": "http://purl.org/ao/",
    "oa": "http://www.w3.org/ns/oa#",
    "dc": "http://purl.org/dc/elements/1.1/",
    "dct": "http://purl.org/dc/terms/",
    "ore": "http://www.openarchives.org/ore/terms/",
    "ro": "http://purl.org/wf4ever/ro#",
    "roterms": "http://purl.org/wf4ever/roterms#",
    "bundle": "http://purl.org/wf4ever/bundle#",
    "prov": "http://www.w3.org/ns/prov#",
    "pav": "http://purl.org/pav/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "foaf": "http://xmlns.com/foaf/0.1/",
    "owl": "http://www.w3.org/2002/07/owl#",

    "uri": "@id",
    "id": {
      "@id": "owl:sameAs",
      "@type": "@id" },
    "file": {
      "@id": "owl:sameAs",
      "@type": "@id" },
    "annotation": {
      "@id": "owl:sameAs",
      "@type": "@id" },

    "manifest": {
        "@id": "ore:isDescribedBy",
        "@type": "@id"
    },

    "createdOn": {
        "@id": "pav:createdOn",
        "@type": "xsd:dateTime"
    },
    "createdBy": {
        "@id": "pav:createdBy",
        "@type": "@id"
    },
    "aggregatedOn": {
        "@id": "pav:createdOn",
        "@type": "xsd:dateTime"
    },
    "aggregatedBy": {
        "@id": "pav:createdBy",
        "@type": "@id"
    },
    "authoredOn": {
        "@id": "pav:authoredOn",
        "@type": "xsd:dateTime"
    },
    "authoredBy": {
        "@id": "pav:authoredBy",
        "@type": "@id"
    },
    "curatedOn": {
        "@id": "pav:curatedOn",
        "@type": "xsd:dateTime"
    },
    "curatedBy": {
        "@id": "pav:curatedBy",
        "@type": "@id"
    },
    "contributedOn": {
        "@id": "pav:contributedOn",
        "@type": "xsd:dateTime"
    },
    "contributedBy": {
        "@id": "pav:contributedBy",
        "@type": "@id"
    },
    "retrievedOn": {
        "@id": "pav:retrievedOn",
```

```
        "@type": "xsd:dateTime"
      },
      "retrievedBy": {
          "@id": "pav:retrievedBy",
          "@type": "@id"
      },
      "retrievedFrom": {
          "@id": "pav:retrievedFrom",
          "@type": "@id"
      },
      "name": {
          "@id": "foaf:name"
      },
      "orcid": {
          "@id": "roterms:orcid",
          "@type": "@id"
      },

      "history": {
          "@id": "prov:has_provenance",
          "@type": "@id"
      },
      "aggregates": {
        "@id": "ore:aggregates",
        "@type": "@id"
      },
      "mediatype": {
          "@id": "dc:format"
      },
      "folder": {
        "@id": "bundle:inFolder",
        "@type": "@id"
      },
      "filename": {
          "@id": "ro:entryName"
      },
      "proxy": {
        "@id": "bundle:hasProxy",
        "@type": "@id"
      },
      "bundledAs": {
          "@id": "bundle:bundledAs",
          "@type": "@id"
      },
      "conformsTo": {
          "@id": "dct:conformsTo",
          "@type": "@id"
      },
      "annotations": {
        "@id": "bundle:hasAnnotation",
        "@type": "@id"
      },
      "content": {
         "@id": "oa:hasBody",
         "@type": "@id"
      },
      "about": {
         "@id": "oa:hasTarget",
         "@type": "@id"
      }

    }
  }
```

3.2.1 Example manifest as triples

As an example of this JSON-LD processing, below is a N-Quads [N-Quads] representation of the triples expressed by `.ro/manifest.json` from the example in section 3.1 .ro/manifest.json, assuming a base URI of `app://2b9486f0-54d8-4274-b241-7669538b0d2f/.ro/manifest.json`

EXAMPLE 4

```
<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.openarchives.org/ore/terms/isDescribedBy> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/.ro/manifest.json> .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://purl.org/pav/createdOn> "2013-03-05T17:29:03Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://purl.org/pav/createdBy> <http://example.com/foaf#alice> .
  <http://example.com/foaf#alice> <http://purl.org/wf4ever/roterms#orcid> <http://orcid.org/0000-0002-1825-0097> .
  <http://example.com/foaf#alice> <http://xmlns.com/foaf/0.1/name> "Alice W. Land" .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.w3.org/ns/prov#has_provenance> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/.ro/evolution.ttl> .


<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.openarchives.org/ore/terms/aggregates> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/folder/soup.jpeg> .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.openarchives.org/ore/terms/aggregates> <http://example.com/blog/> .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.openarchives.org/ore/terms/aggregates> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/README.txt> .
  <app://2b9486f0-54d8-4274-b241-7669538b0d2f/README.txt> <http://purl.org/dc/elements/1.1/format> "text/plain" .
  <app://2b9486f0-54d8-4274-b241-7669538b0d2f/README.txt> <http://purl.org/pav/createdOn> "2013-02-12T19:37:32.939Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
  <app://2b9486f0-54d8-4274-b241-7669538b0d2f/README.txt> <http://purl.org/pav/createdBy> <http://example.com/foaf#bob> .
    <http://example.com/foaf#bob> <http://xmlns.com/foaf/0.1/name> "Bob Builder" .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://www.openarchives.org/ore/terms/aggregates> <http://example.com/comments.txt> .

  _:b2 <http://purl.org/wf4ever/bundle#hasProxy> <urn:uuid:a0cf8616-bee4-4a71-b21e-c60e6499a644> .
  _:b2 <http://purl.org/wf4ever/bundle#inFolder> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/folder/> .
  _:b2 <http://purl.org/wf4ever/ro#entryName> "external.txt" .
  _:b2 <http://www.openarchives.org/ore/terms/proxyFor> <http://example.com/comments.txt> .


<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://purl.org/wf4ever/bundle#hasAnnotation> <urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf> .
  <urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf> <http://www.w3.org/ns/oa#hasBody> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/.ro/annotations/soup-properties.ttl> .
  <urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf> <http://www.w3.org/ns/oa#hasTarget> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/folder/soup.jpeg> .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://purl.org/wf4ever/bundle#hasAnnotation> _:b0 .
  _:b0 <http://www.w3.org/ns/oa#hasBody> <http://example.com/blog/they-aggregated-our-file> .
  _:b0 <http://www.w3.org/ns/oa#hasTarget> <urn:uuid:a0cf8616-bee4-4a71-b21e-c60e6499a644> .

<app://2b9486f0-54d8-4274-b241-7669538b0d2f/> <http://purl.org/wf4ever/bundle#hasAnnotation> _:b1 .
  _:b1 <http://www.w3.org/ns/oa#hasBody> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/.ro/annotations/a-meta-annotation-in-this-ro.txt> .
  _:b1 <http://www.w3.org/ns/oa#hasTarget> <app://2b9486f0-54d8-4274-b241-7669538b0d2f/> .
  _:b1 <http://www.w3.org/ns/oa#hasTarget> <urn:uuid:d67466b4-3aeb-4855-8203-90febe71abdf> .
```

3.3 Custom JSON-LD

Applications who support JSON-LD (rather than just JSON) MAY choose to parse and generate statements in `.ro/manifest.json` according to the [JSON-LD]

specifications.

Applications generating JSON-LD MAY provide additional items in the `@context` list, but SHOULD include `https://w3id.org/bundle/context` as the last item, to indicate to JSON parsers that the manifest can be parsed as plain JSON according to section 3.1 .ro/manifest.json. Applications SHOULD NOT use `@context` at deeper nexting levels.

Applications MAY add other properties directly to JSON Objects defined from section 3.1 .ro/manifest.json, but MUST ensure they are valid JSON-LD. Note that the RO Bundle JSON-LD context does not specify the typing of properties outside this specification. As an example, to provide additional [FOAF] properties for the creator of a file:

EXAMPLE 5

```
{
    "@context":  [
        "https://w3id.org/bundle/context"
    ],
    "id": "/",
    "manifest": "manifest.json",
    "aggregates": [
       { "uri":      "/README.txt",
         "createdBy": {
             "uri":     "http://example.com/foaf#bob",
             "name":    "Bob Builder",

             "foaf:homepage": {
                 "@id": "http://example.com/bob"
             },
             "foaf:title": "Dr"

         }
      }
    ]
}
```

Alternatively, additional statements can be made within a top-level `@graph` node according to JSON-LD Named Graphs. For example:

EXAMPLE 6

```
{
    "@context":  [
        "https://w3id.org/bundle/context"
    ],
    "id": "/
    "manifest": "manifest.json",
    "aggregates": [
       { "uri": "http://example.com/blog/2012" },
       { "uri": "http://example.com/blog/2013" }
    ],

    "@graph": [
      { "@id": "http://example.com/blog/2013",
        "dcterms:replaces": "http://example.com/blog/2012" },
      { "@id": "http://example.com/blog/2012",
        "dcterms:isReplacedBy": "http://example.com/blog/2013" }
    ]

}
```

Note that rather than using the above extension mechanism, it is generally RECOMMENDED to instead store such additional statements in an annotation content for purposes of provenance and separation of concern. Although technically valid, it is NOT RECOMMENDED to use the member `@graph` to embed semantic annotation bodies within `annotations` nodes, as it would duplicate the content of the annotation body in the bundle and may lead to inconsistencies.

3.4 Alternative manifest representations

In addition to the `.ro/manifest.json` representation specified in section 3.1 .ro/manifest.json, a Research Object Bundle MAY include the ORE manifest in alternative representations like RDF/XML [RDF-SYNTAX-GRAMMAR] and Turtle [TURTLE], for instance by generating them using the conversion from JSON to RDF algorithm in JSON-LD API [JSON-LD].

- Alternative manifests SHOULD have a path starting with `.ro/manifest`, for instance `.ro/manifest.ttl` for a Turtle representation.
- When multiple manifests are present, applications SHOULD consider `.ro/manifest.json` as the authorative representation of the research object.
- Alternative manifests SHOULD represent (at least) the equivalent RDF graph of `.ro/manifest.json` (see section 3.2 JSON-LD and mapping to RO model)
- Alternative manifests SHOULD be listed in the `META-INF/container.xml` as `<rootfile>` entries with corresponding `media-type` attributes.
- Any alternative manifest listed as a *rootfile* MUST minimally represent the same conceptual information as `.ro/manifest.json`

If an application is modifying a research object bundle which contains manifests it can't handle (and thus can't update), the application SHOULD remove the `rootfile` entry for those unsupported manifests, and MAY delete those manifests from the archive.

4. Identifiers

This section is non-normative.

Objects in a research object bundle are identified within the JSON manifest relative to `.ro/manifest.json`, with `/` resolving to the root of the ZIP archive.

Prefix	Interpretation
`/`	Path relative to root of ZIP archive
`urn:uuid:`	UUID according to [RFC4122]
(containing `:`)	Absolute URI
(no prefix)	Path relative to `/.ro/` in the ZIP archive

Due to their nature as ZIP files, Research Object Bundles might be downloaded, copied, moved and republished. In order to avoid ambiguity about RO identity and evolution, each Research Object Bundle serialization is considered to represent unique Research Objects. Thus any of the prefixes above describing resources within the bundle are relative to the root of the ZIP file, and the `id` identifying the Research Object is set to `"/"`, meaning the root represents the RO itself.

4.1 Escaping URIs

All identifieres within the manifest are in JSON expressed as escaped [IRI]s. If the ZIP file contains a filename with escapable characters, e.g. `/folder with spaces/Δfilename-€unicode.txt`, then this SHOULD in JSON be expressed as an escaped IRI, e.g. `http://example.com/folder%20with%20spaces/Δfilename-€unicode`, but MAY alternatively be expressed as an ASCII-escaped URI, e.g. `/folder%20with%20spaces/%CE%94filename-%E2%88%88unicode.txt`.

If the research object references an absolute URI (e.g. aggregating or as an annotation body), e.g. `http://www.example.中國/with%20space%20as%20well.txt`, then this SHOULD in JSON be expressed as an IRI as-is (preserving IRI escapes for special characters), or using the [RFC3492] punycode DNS name: `http://xn--`

`fiqs8s.example.com`.

The filenames as stored in the ZIP file MUST be a valid UTF-8 filename and SHOULD NOT be URI escaped, as this causes double escaping within the manifest.

Note that IRIs that syntactically differ may be identifying the same resource if they match after comparing them as absolute IRIs within the bundle.

4.2 Absolute URIs for bundle resources

Applications which require an absolute URI for identifying a resource within a Research Object Bundle may choose to use the approach presented in this section in combination with resolving against the prefix table above.

The app: URI scheme [[APP-URI!]] proposes how a URI can be formed for the purposes of accessing resources within a ZIP file as if the resources were retrieved from a HTTP server. While this is primary intended for sandboxing HTML applications, it is equally applicable to Research Object bundles for the purposes of sandboxing and for generating a URI independent of the location of the ZIP archive.

The app: URI scheme suggests generating a UUID string [RFC4122] for minting the *authority*, forming the *base URI* for the RO bundle. For instance, if:

```
http://example.com/example1.robundle
```

contains the file `/folder/helloworld.txt`, then we generate a new UUID `8191dee8-0b8e-452d-8d64-7706a140185e` and refer to the Research Object as

```
app://8191dee8-0b8e-452d-8d64-7706a140185e/
```

and can refer to its bundled file `/folder/helloworld.txt` as:

```
app://8191dee8-0b8e-452d-8d64-7706a140185e/folder/helloworld.txt
```

The type of authority to generate depends on what is the purpose of the absolute URI:

1. For security/sandboxing when interpreting RO bundles, the authority should be a v4 UUID from random numbers. Thus the URI is guaranteed to be unique for each interpretation, and can't (reasonably) be pre-guessed.
   Applications exposing such URIs might want to record the provenance using the pav:retrievedFrom relation to indicate where the bundle was retrieved from. For instance (in Turtle):

   EXAMPLE 7

   ```
   @prefix pav: <http://purl.org/pav/> .
   @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
   <app://15259726-dcbb-42ff-8fc6-36282c98d4e6/>
     pav:retrievedFrom <http://example.com/example1.robundle> ;
     pav:retrievedOn "2013-05-21T14:24:19Z"^^xsd:dateTime .
   ```

2. For describing/referencing content of an RO bundle accessed at a given URL, the authority should be generated as a name based UUID using v5 (SHA-1 hashing) concatination of the URL namespace `6ba7b811-9dad-11d1-80b4-00c04fd430c8` (as UUID bytes) and the ASCII-escaped version of the URL. This approach gives a predictable UUID for a particular URL, even if the content at the URL might later change.
   Applications using this approach might want to declare the equivalent of a `owl:sameAs` relation between the accessed URI and the generated `app:` URI in order to record the original URI. For instance:

   EXAMPLE 8

   ```
   @prefix owl: <http://www.w3.org/2002/07/owl#> .
   <app://7878e885-327c-5ad4-9868-7338f1f13b3b/> owl:sameAs
         <http://example.com/example1.robundle> .
   ```

3. For purposes of describing the content of an RO bundle as a bytestream independent of its location (for instance on a USB stick), then the authority should be the hexadecimal SHA-256 checksum of the ZIP archive (not a UUID). This ensures a predictable URI for the same physical representation, where any change to the bundle generates a new identifier.
   Applications exposing such URIs might want to record the equivalent provenance of a pav:retrievedFrom relation to indicate where the bundle was retrieved from, including the time of retrieval using the equivalent of pav:retrievedOn. (See example above).

Example app base URIs:

- `app://15259726-dcbb-42ff-8fc6-36282c98d4e6/` *UUID v4 using pseduo-random number*
- `app://7878e885-327c-5ad4-9868-7338f1f13b3b/` *UUID v5 of the URL* `http://example.com/bundle1.robundle`
- `app://587cff3ae37d58af6886d656623bd91237759a42d8fe1575a9744898c01d97d7/` *SHA-256 of an empty RO bundle*

5. Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words MAY, MUST, MUST NOT, NOT RECOMMENDED, OPTIONAL, RECOMMENDED, SHOULD, and SHOULD NOT are to be interpreted as described in [RFC2119].

A. Changes

2014-11-05

- Pubilsher is researchobject.org instead of Wf4Ever
- Changes to be suggested on Github instead of mailing list
- URI vs IRI
- Added section for Escaping URIs
- Rootfile section is now normative
- `.ro` MUST be present
- `.ro/manifest.json` MUST be present
- Common provenance keys (`createdOn`, `authoredOn`, ..) now in separate section
- Added `retrievedFrom`, `retrievedOn` and `retrievedBy`
- Added `conformsTo`
- Keys `file`, `proxy`, `annotation` replaced with `uri`
- Annotations are required to relate to a bundle resource
- Detailed mapping to absolute URIs with examples
- Example translation now in triples rather than Turtle
- Now allowed to add custom JSON-LD to any node
- More references made normative
- Updated reference for ROAPI

B. Acknowledgements

Thanks to Khalid Belhajjame, Graham Klyne and Piotr Hołubowicz for reviewing this specification. The underlying work has been funded as part of the Wf4Ever project, funded by the European Commisson's FP7 programme (FP7-ICT-2007-6 270192). Many thanks to Robin Berjon and contributors for making ReSpec.js which generated this page.

C. References

C.1 Normative references

[IRI]
M. Duerst; M. Suignard. *Internationalized Resource Identifiers (IRIs)*. January 2005. Proposed Standard. URL: http://www.ietf.org/rfc/rfc3987.txt

[JSON-LD]
Manu Sporny; Gregg Kellogg; Markus Lanthaler. *JSON-LD 1.0*. 16 January 2014. W3C Recommendation. URL: http://www.w3.org/TR/json-ld/

[OA]
Robert Sanderson; Paolo Ciccarese; Herbert Van de Sompel. *Open Annotation Data Model, Community Draft, 08 February 2013*. 8 February 2013. URL: http://www.openannotation.org/spec/core/

[ORE]
Open Archives Initiative Object Reuse and Exchange. *ORE Specifications and User Guides*. URL: http://www.openarchives.org/ore/1.0/toc.html

[RFC2119]
S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC4122]
P. Leach; M. Mealling; R. Salz. *A Universally Unique IDentifier (UUID) URN Namespace*. July 2005. Proposed Standard. URL: http://www.ietf.org/rfc/rfc4122.txt

[RFC4627]
D. Crockford. *The application/json Media Type for JavaScript Object Notation (JSON)*. July 2006. Informational. URL: http://www.ietf.org/rfc/rfc4627.txt

[RO]
Stian Soiland-Reyes; Sean Bechhofer et al.: *Wf4Ever Research Object Model*. 30 November 2013. URL: http://purl.org/wf4ever/model

[UCF]
Adobe: *Universal Container Format*. URL: https://learn.adobe.com/wiki/display/PDFNAV/Universal+Container+Format accessed 2013-02-28.

[URI]
T. Berners-Lee; R. Fielding; L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. January 2005. Internet Standard. URL: http://www.ietf.org/rfc/rfc3986.txt

[ZIP]
.ZIP File Format Specification. 1 September 2012. Final. URL: http://www.pkware.com/documents/casestudies/APPNOTE.TXT

C.2 Informative references

[FOAF]
Dan Brickley; Libby Miller. *FOAF Vocabulary Specification 0.99 (Paddington Edition)*. 14 January 2014. URL: http://xmlns.com/foaf/spec

[N-Quads]
Gavin Carothers. *RDF 1.1 N-Quads*. 25 February 2014. W3C Recommendation. URL: http://www.w3.org/TR/n-quads/

[OCF]
James Pritchett; Markus Gylling: *EPUB Open Container Format (OCF) 3.0*. *Recommended Specification 11 October 2011. International Digital Publishing Forum (IDPF). URL: http://idpf.org/epub/30/spec/epub30-ocf-20111011.html*

[ODF]
Rob Weir; Michael Brauer; Patrick Durusau; Dennis Hamilton: Open Document Format for Office Applications (OpenDocument) Version 1.3, Part 3: Packages. OASIS Standard. 29 September 2011. URL: http://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2-part3.html

[RDF-SYNTAX-GRAMMAR]
Fabien Gandon; Guus Schreiber. *RDF 1.1 XML Syntax*. 25 February 2014. W3C Recommendation. URL: http://www.w3.org/TR/rdf-syntax-grammar/

[RFC3492]
A. Costello. *Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)*. March 2003. Proposed Standard. URL: http://www.ietf.org/rfc/rfc3492.txt

[RFC6839]
T. Hansen; A. Melnikov. *Additional Media Type Structured Syntax Suffixes*. January 2013. Informational. URL: http://www.ietf.org/rfc/rfc6839.txt

[ROAPI]
Piotr Holubowicz; Graham Klyne; Raúl Palma; Stian Soiland-Reyes; Filip Wiśniewski. *RO API 6*. URL: http://www.wf4ever-project.org/wiki/display/docs/RO+API+6

[ROPrimer]
Jun Zhao et al: *Wf4Ever Research Object Ontologies and Vocabularies Primer*. 13 April 2012. URL: http://purl.org/wf4ever/primer accessed 2013-02-27

[TURTLE]
Eric Prud'hommeaux; Gavin Carothers. *RDF 1.1 Turtle*. 25 February 2014. W3C Recommendation. URL: http://www.w3.org/TR/turtle/