

Parties and Locations

Individuals, teams, corporations and government agencies are represented in OSCAL `metadata` using the `parties` array. Location information can be included within a party's information or defined separately for sharing.

Locations

Define a common location to be associated with multiple parties, or as stand-alone information. In `metadata` include:

- a `locations` array. Each entry has:
 - a `uuid` (required)
 - a `title` (best practice)
 - an `address` object with:
 - a `type` set to `work`
 - an `address-lines` array with one or more string entries representing the street number, mail stop, or similar.
 - a `city` as appropriate for your geography
 - a `state` as appropriate for your geography
 - a `postal-code` as appropriate for your geography

```
root-model-name
  metadata:

    locations:
      - uuid: 11111111-2222-4000-8000-003000000001
        title: CSP HQ
        address:
          type: work
          addr-lines:
            - Suite 0000
            - 1234 Some Street
          city: Haven
          state: ME
          postal-code: '00000'
```

Locations for Stand-Alone Information

For locations such as Data Centers that may be expressed in an SSP, POA&M, AP or AR:

- add a `props` array with an entry:
 - `name` set to `type`
 - `value` set to `data-center` (Additional values are allowed, but are not well-defined at this time.)
 - `class` set to either `primary` or `alternate`

```

root-model-name
  metadata:

    locations:
      - uuid: 11111111-2222-4000-8000-003000000002
        title: Primary Data Center
        address:
          addr-lines:
            - 2222 Main Street
          city: Anywhere
          state: --
          postal-code: 00000-0000
          country: US
        props:
          - name: type
            value: data-center
            class: primary

```

Parties

In `metadata` include:

- a `parties` array. Each entry has:
 - a `uuid` (required)
 - a `type` (required) set to `individual` or `organization`
 - Use `organization` for teams, companies and agencies.
 - a `name` (best practice)
 - a `short-name` (optional - recommended for organizations with well-known acronyms)
 - an `email-addresses` array with each entry containing
 - a string representing an [RFC-6531](#) formatted email address
 - a `telephone-numbers` array with each entry containing an object with:
 - `type` (optional) set to `home`, `office` or `mobile` (Other values allowed, but not well defined)
 - `number` set to a string representing the phone number
 - **either** a `location-uuids` array or an `addresses` array:

- `location-uuids` array entries contain the UUID value of defined `locations` array entries.
- `addresses` array entries each include:
 - a `type` set to `work` or `home` as appropriate
 - an `address-lines` array with one or more string entries representing the street number, mail stop, or similar.
 - a `city` as appropriate for your geography
 - a `state` as appropriate for your geography
 - a `postal-code` as appropriate for your geography

```

root-model-name
  metadata:

  parties:
  - uuid: 11111111-2222-4000-8000-004000000001
    type: organization
    name: Cloud Service Provider (CSP) Name
    short-name: CSP Acronym/Short Name
    email-addresses:
    - name@example.com
    telephone-numbers:
    - number: '2020000001':
    location-uuids:
    - 11111111-2222-4000-8000-003000000001

```

Alternatively use `addresses` instead of `location-uuids`:

```

addresses:
- type: work
  addr-lines:
  - 1800 F St. NW
  city: Washington
  state: DC
  postal-code: '20006'
  country: US

```

Logos and Web Sties

To associate a logo or web site with a party:

- add a `links` array to the party

To identify a web site:

- add an entry to the `links` array with:
 - `href` set to the URL of the organization's web site
 - `rel` set to `homepage`

To identify a logo:

- add an entry to the `back-matter` / `resources` array with:
 - a `uuid` (required)
 - a `title` (best practice)
 - `description` (optional)
 - add a `props` array with an entry:
 - `name` set to `type`
 - `value` set to `logo`
 - **either** an `rlinks` array or a `base64` object
 - an `rlinks` entry includes:
 - `href` (required) with a path to the logo (relative path strongly recommended)
 - `media-type` (best practice) with an appropriate [IANA-recognized Media Type](#)
 - a `base64` object includes:
 - `media-type` (required for rendering) with an appropriate [IANA-recognized Media Type](#)
 - `value` with the Base 64 value of the logo
- add an entry to the `links` array with:
 - `href` with a URI Fragment that references the UUID of the `resources`
 - The `href` value must start with a hashtag (`#`) character followed by the UUID value of the resource
 - `rel` set to `logo`

```
root-model-name
  metadata:

  parties:
    - uuid: d865602c-9d3b-49d7-8125-ce3f1ca04231
      type: organization
      name: CSP
```

links:

- href: <https://csp.example.com>
rel: homepage
- href: #891263fb-a5d6-44db-8d73-51bb8a9a3610
rel: logo

back-matter:

resources:

- uuid: 891263fb-a5d6-44db-8d73-51bb8a9a3610
title: CSP Logo
description: Logo of the organization that prepared the document.

props:

- name: type
value: logo

rlinks:

- href: ./attachments/img/logo.png

base64:

- filename: logo.png
media-type: image/png
value: 00000000

Note: For the logo, use rlink with a relative path or embed the logo as base64.

Revision #1

Created 2026-04-08 21:05:46 UTC by Brian Ruf

Updated 2026-04-08 22:09:57 UTC by Brian Ruf