
collective.jsonmigrator Documentation

Release 0.1.1

garbas

Jun 11, 2018

Contents

1	<code>collective.jsonmigrator.jsonsource</code>	3
1.1	Parameters	3
1.2	Example	3
2	<code>collective.jsonmigrator.skipitems</code>	5
2.1	Parameters	5
2.2	Example	5
3	<code>collective.jsonmigrator.partialcommit</code>	7
3.1	Parameters	7
3.2	Example	7
4	<code>collective.jsonmigrator.statistics</code>	9
4.1	Parameters	9
4.2	Example	9
5	<code>collective.jsonmigrator.workflowhistory</code>	11
5.1	Parameters	11
5.2	Example	11
6	<code>collective.jsonmigrator.mimetype</code>	13
6.1	Parameters	13
6.2	Example	13
7	<code>collective.jsonmigrator.properties</code>	15
7.1	Configuration options	15
7.2	Example	16
8	<code>collective.jsonmigrator.permission_mapping</code>	17
8.1	Parameters	17
8.2	Example	17
9	<code>collective.jsonmigrator.owner</code>	19
9.1	Parameters	19
9.2	Example	19
10	<code>collective.jsonmigrator.local_roles</code>	21
10.1	Parameters	21

10.2 Example	21
11 collective.jsonmigrator.datafields	23
11.1 Configuration options	23
11.2 Example	23

The purpose of the `collective.jsonmigrator` package is to provide a set of blueprints that help you to migrate content into [Plone](#). (blueprint is an extension to [collective.transmogrifier](#)).

It provides a list of blueprints built around `collective.jsonmigrator.jsonsource` with the purpose of providing flexible infrastructure to do migrations in Plone.

And just incase you forgot, migration is a bitch... so have fun :P

CHAPTER 1

`collective.jsonmigrator.jsonsource`

Read JSON files and insert them into transmogrifier pipeline.

1.1 Parameters

path (required) Path to directory containing JSON files (look in example below).

Also possible to specify in `some.package:path/to/json/directory` way.

1.2 Example

Configuration:

```
[transmogrifier]
pipeline =
  source

[source]
blueprint = collective.jsonmigrator.jsonsource
path = some.package:/path/to/json/dir
```

JSON files structure:

```
some.package:/path/to/json/dir
|-> 0/
    |-> 1.json
    |-> 2.json
    ...
    |-> 999.json
|-> 1/
    |-> 1000.json
```

(continues on next page)

(continued from previous page)

```
| -> 1001.json  
...
```

JSON file:

```
{  
  "_path": "/Plone/front-page",  
  "_type": "Document",  
  ...  
}
```

```
collective.jsonmigrator.skipitems
```

Skip first N item in pipeline.

Development blueprint. Useful when you are processing big data pipelines and you know that the first N items are already migrated.

2.1 Parameters

first (required) define number of items from the beginning of data pipeline to skip.

2.2 Example

Configuration:

```
[transmogrifier]
pipeline =
  source
  skipitems

...

[skipitems]
blueprint = collective.jsonmigrator.skipitems
first = 10000
```

```
collective.jsonmigrator.partialcommit
```

Used to commit after some items have been processed.

3.1 Parameters

every (default 100): Define number of items after which commit (writing to ZODB) will happen.

3.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    commit

...

[commit]
blueprint = collective.jsonmigrator.partialcommit
every = 500
```

```
collective.jsonmigrator.statistics
```

TODO need to fix statistic blueprint so it doesn't depend on other blueprints to report and add statistic data.

Also reporting should not only be written to `stdout`.

4.1 Parameters

4.2 Example

`collective.jsonmigrator.workflowhistory`

Update the workflow history of an object.

5.1 Parameters

No parameters.

Expected data in pipeline:

- **_path**: path to object on which we want to change workflow history.
- **_workflow_history**: workflow history to be applied to object resolved above.

5.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    workflowhistory

...

[workflowhistory]
blueprint = collective.jsonmigrator.workflowhistory
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_workflow_history": {
```

(continues on next page)

(continued from previous page)

```
    "plone_workflow": [  
      {  
        "action": null,  
        "review_state": "visible",  
        "comments": "",  
        "actor": "admin",  
        "time": "2010/09/15 02:19:57.932 GMT+2"  
      }  
    ],  
  },  
}
```

```
collective.jsonmigrator.mimetype
```

Sometimes we need to fix/change the mimetype of migrated objects.

6.1 Parameters

No parameters.

Expected data in pipeline:

- **_path**: path to object on which we want to change mimetype.
- **_content_type**: mimetype to be applied to object resolved above.

6.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    mimetype
...

[mimetype]
blueprint = collective.jsonmigrator.mimetype
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_content_type": "text/html",
}
```

`collective.jsonmigrator.properties`

Update properties of an object.

7.1 Configuration options

No specific blueprint parameters.

Expected data structure in pipeline:

- **_path**: path to object on which we want to change properties.
- **_properties**: properties to be applied to object resolved above.

properties passed in this data field (as shown in example) is a list of 3-item lists.:

```
[
  [
    <property-name>,
    <property-value>,
    <property-type>
  ],
  [
    <property2-name>,
    <property2-value>,
    <property2-type>
  ],
  ...
]
```

<property-type> is set of types which you can select through the ZMI when you edit/add a property.

7.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    properties
...

[properties]
blueprint = collective.jsonmigrator.properties
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_properties": [
    [
      "title",
      "Welcome to Plone",
      "string"
    ]
  ],
}
```

`collective.jsonmigrator.permission_mapping`

Update permissions of an object.

8.1 Parameters

No parameters.

Expected data in pipeline:

- **_path**: path to object on which we want to change permissions.
- **_permission_mapping**: permissions to be applied to object resolved above.

8.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    permission_mapping
...

[mimetype]
blueprint = collective.jsonmigrator.permission_mapping
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_permission_mapping": {
```

(continues on next page)

(continued from previous page)

```
"Modify portal content": {
    "acquire": false,
    "roles": [
        "Manager",
        "Owner"
    ]
},
"Access contents information": {
    "acquire": true,
    "roles": [
        "Anonymous",
        "Manager",
        "Reviewer"
    ]
},
"View": {
    "acquire": true,
    "roles": [
        "Anonymous",
        "Manager",
        "Reviewer"
    ]
}
},
}
```

`collective.jsonmigrator.owner`

Update owner of an object.

9.1 Parameters

No parameters.

Expected data in pipeline:

- **_path**: path to object on which we want to change properties.
- **_owner**: properties to be applied to object resolved above.

9.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    owner
...

[owner]
blueprint = collective.jsonmigrator.owner
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_owner": [
```

(continues on next page)

(continued from previous page)

```
    1,  
    "admin"  
  ],  
}
```


CHAPTER 10

`collective.jsonmigrator.local_roles`

Update local roles of an object.

10.1 Parameters

No parameters.

- **_path**: path to object on which we want to change local roles.
- **_ac_local_roles**: local roles to be applied to object resolved above.

10.2 Example

Configuration:

```
[transmogrifier]
pipeline =
    source
    local_roles

...

[local_roles]
blueprint = collective.jsonmigrator.local_roles
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_ac_local_roles": {
    "admin": [
      "Owner"
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```
    ],  
    },  
}
```

```
collective.jsonmigrator.datafields
```

Update data/blob fields of an object.

TODO missing base path (maybe even passed somehow from source blueprint)

TODO only update if needed

11.1 Configuration options

No specific blueprint parameters.

Expected data structure in pipeline:

- **_path**: path to object on which we want to change local roles.
- **_datafield_<field>**: field which needs to store data

11.2 Example

This example will try to store content of `0/1.json-file-1` into the `attachment` field of the `/Plone/index_html` object.

Configuration:

```
[transmogrifier]
pipeline =
    source
    datafields

...

[datafields]
blueprint = collective.jsonmigrator.datafields
```

Data in pipeline:

```
{
  "_path": "/Plone/index_html",
  "_datafield_attachment": "0/1.json-file-1",
}
```