



# Rosetta SIP Factory

Python library for building Submission  
Information Packages for the Rosetta  
digital preservation application

National Library of New Zealand

# Rosetta SIP Factory

- What is it?
- What does it do?
- Why was it made?
- How does it work?



# Rosetta SIP Factory: What is it?

- The SIP Factory is a Python library.
- It helps you make Submission Information Packages.
- More specifically, it helps you make SIPs for the Rosetta Digital Preservation application.



# Rosetta SIP Factory: What is it?

- Rosetta: So what's that?
- It's a commercial application developed by Ex Libris (A ProQuest Company)
- National Library of New Zealand was a development partner of Rosetta
- It uses METS to structure the descriptive and technical metadata for SIPs, AIPs and DIPs
- Uses the dmdSec, amdSec, fileSec and structMap METS sections
- Implements a flavour of PREMIS called DNX



# Rosetta SIP Factory: What does it do?

- It builds SIPs (or more to the point, helps you build SIPs)...
- ...but only if you're happy to do a little coding!
- Built to accommodate edge cases in NLNZ's digital collection and preservation workflows
- Intended to have a relatively low barrier to entry...
- ...but does not have "training wheels" built in!



# Rosetta SIP Factory: What does it do?

- The sip\_builder module has three functions:
  - build\_sip()
  - build\_single\_file\_sip()
  - build\_sip\_from\_json()
- They all take Dublin Core metadata, and they have parameters for sections of DNX metadata.
- build\_single\_file\_sip takes a single filepath;
- build\_sip takes directory paths for pres master, modified master, and access derivative (only pres master is mandatory);
- build\_sip\_from\_json takes a json document with DNX values, filepaths, checksums, etc.



# Rosetta SIP Factory: Why was it built?

- It was built for edge cases!
- NLNZ is quietly fostering a new generation of technically-savvy librarians, who are quite keen on Python.
- They wanted a code library that would allow them to implement custom business logic, without the technical and resourcing overheads of conventional dev work
- Offers staff a chance to explore their own ideas about new workflows



# Rosetta SIP Factory: How does it work?

```
from rosetta_sip_factory import sip_builder
sip_builder.build_sip(
    ie_dmd_dict=[{'dc:title': 'Title of IE'}],
    input_dir="/path/to/ie",
    pres_master_dir="/path/to/ie/pm",
    general_ie_characteristics=[{"IEEntityType":
        "OneOffIE"}],
    access_rights_policy=[{"policyId": "1001"}],
    output_dir="/path/to/output_sip"
)
```





# Rosetta SIP Factory

## Repository:

[https://github.com/NLNZDigitalPreservation/rosetta\\_sip\\_factory](https://github.com/NLNZDigitalPreservation/rosetta_sip_factory)

## Python Package Index (pip):

```
pip install rosetta_sip_factory
```

