

# Options for producing OpenMC h5 cross sections

IAEA Consultancy Meeting on Model Code Output  
&  
Application Nuclear Data Form Structure

# Obtaining h5 cross section files

- Download compressed (tar) libraries from [OpenMC.org](https://openmc.org)
- Download individual isotopes using the openmc-data-downloader Python package [https://github.com/openmc-data-storage/openmc\\_data\\_downloader](https://github.com/openmc-data-storage/openmc_data_downloader)
- Process h5 files using the data repository <https://github.com/openmc-dev/data>
  - Process ACE files using `convert*.py` scripts
  - Process ENDF files using `generate*.py` scripts

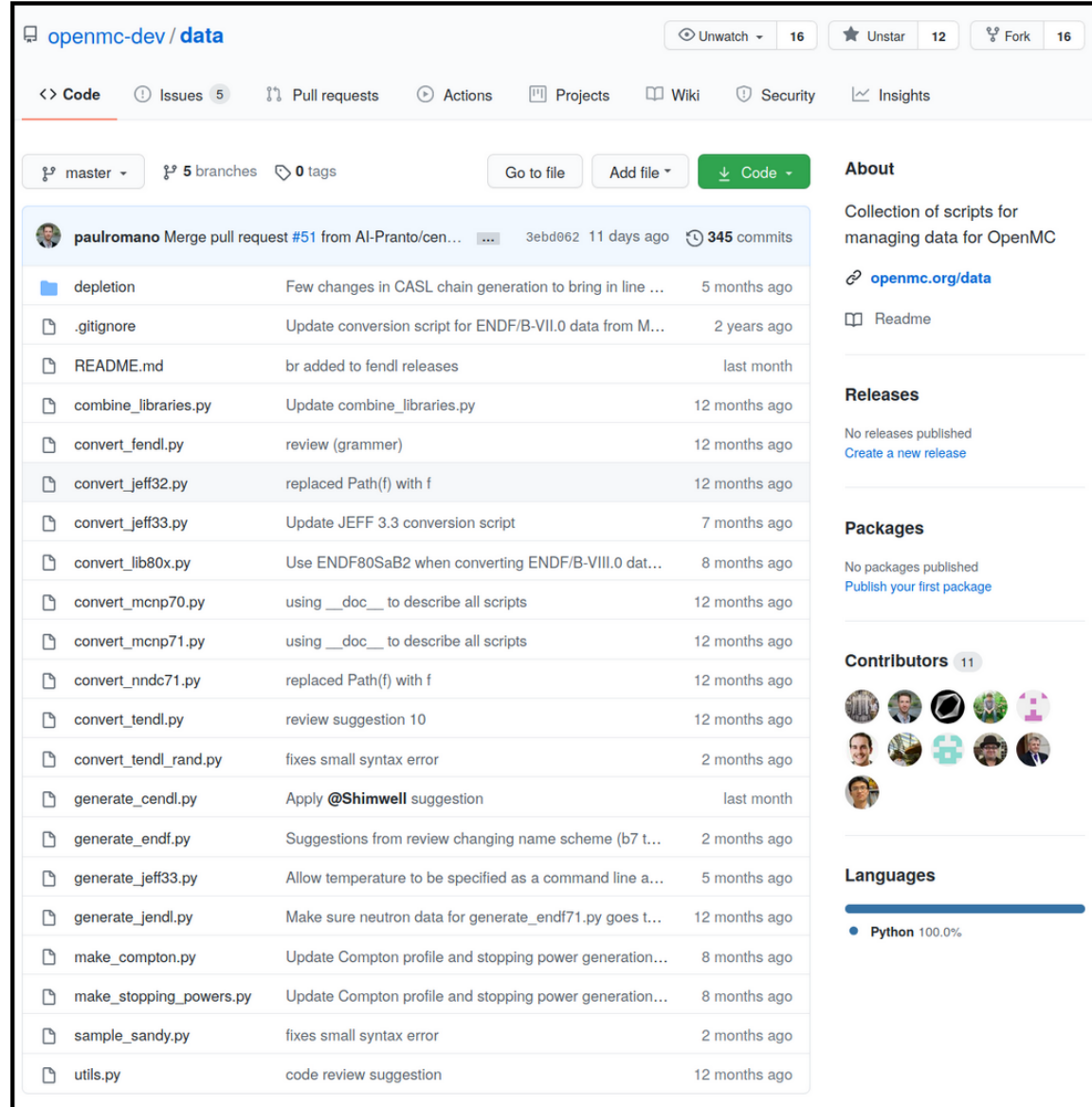
# Obtaining h5 cross section files

There are several ways of obtaining the cross section data in the h5 file format required for neutronics simulations.

Library	Release	Processed by	Download from openmc.org	Download from openmc-data-downloader	Download ACE files and convert HDF5	Download ENDF files and generate HDF5	Convert local ACE files
CENDL	3.1 3.2					generate_cendl.py	
ENDF/B	VII.0	LANL	✓				convert_mcnp70.py
ENDF/B	VII.1	LANL	✓				convert_mcnp71.py
ENDF/B	VII.1	NNDC	✓	✓	convert_nndc71.py	generate_endf.py	
ENDF/B	VIII.0	LANL	✓				convert_lib80x.py
ENDF/B	VIII.0	NNDC	✓				
FENDL	2.1, 3.0, 3.1a, 3.1d			✓ (3.1d)	convert_fendl.py		
JENDL	4.0					generate_jendl.py	
JEFF	3.2		✓		convert_jeff32.py		
JEFF	3.3		✓		convert_jeff33.py		
TENDL	2015, 2017, 2019			✓ (2019)	convert_tendl.py		

# OpenMC data repository

- Python scripts
- Permissive open source license (MIT)
- 11 contributors
- Storage and sharing opportunity to avoid reproducing the same scripts
- Useful for developers
- Generate scripts require NJOY install
- PyPi package with pip install on the horizon



openmc-dev / data

Unwatch 16 Unstar 12 Fork 16

<> Code Issues 5 Pull requests Actions Projects Wiki Security Insights

master 5 branches 0 tags Go to file Add file Code

paulromano Merge pull request #51 from AI-Pranto/cen... 3ebd062 11 days ago 345 commits

depletion	Few changes in CASL chain generation to bring in line ...	5 months ago
.gitignore	Update conversion script for ENDF/B-VII.0 data from M...	2 years ago
README.md	br added to fendl releases	last month
combine_libraries.py	Update combine_libraries.py	12 months ago
convert_fendl.py	review (grammer)	12 months ago
convert_jeff32.py	replaced Path(f) with f	12 months ago
convert_jeff33.py	Update JEFF 3.3 conversion script	7 months ago
convert_lib80x.py	Use ENDF80SaB2 when converting ENDF/B-VIII.0 dat...	8 months ago
convert_mcnp70.py	using __doc__ to describe all scripts	12 months ago
convert_mcnp71.py	using __doc__ to describe all scripts	12 months ago
convert_nndc71.py	replaced Path(f) with f	12 months ago
convert_tendl.py	review suggestion 10	12 months ago
convert_tendl_rand.py	fixes small syntax error	2 months ago
generate_cendl.py	Apply @Shimwell suggestion	last month
generate_endf.py	Suggestions from review changing name scheme (b7 t...	2 months ago
generate_jeff33.py	Allow temperature to be specified as a command line a...	5 months ago
generate_jendl.py	Make sure neutron data for generate_endf71.py goes t...	12 months ago
make_compton.py	Update Compton profile and stopping power generation...	8 months ago
make_stopping_powers.py	Update Compton profile and stopping power generation...	8 months ago
sample_sandy.py	fixes small syntax error	2 months ago
utils.py	code review suggestion	12 months ago

About

Collection of scripts for managing data for OpenMC

[openmc.org/data](https://openmc.org/data)

Readme

Releases

No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)

Contributors 11

Languages

- Python 100.0%

# Obtaining h5 cross sections - data repo

- Example usage of one of the Python scripts
- This particular script downloads 429MB of compressed ACE and ENDF files
- Extracts the files
- Processes the files into h5 file for use in OpenMC

```
git clone https://github.com/openmc-dev/data.git

cd data

python convert_fendl.py --release 3.1d
>> downloading, extracting and processing ...

ls
>> fendl-3.1d-ace/
>> fendl-3.1d-endf/
>> fendl-3.1d-download/
>> fendl-3.1d-hdf5/

ls fendl-3.1d-hdf5/
>> cross_sections.xml
>> neutron/
>> photon/

ls fendl-3.1d-hdf5/neutron
>> Ag107.h5
>> Ag109.h5
>> Al27.h5
>> Ar36.h5
>> ...

export OPENMC_CROSS_SECTIONS=fendl-3.1d-hdf5/cross
```

# OpenMC.org

- <https://openmc.org> includes a data section with downloads
  - Official Libraries
  - LANL-Based Data Libraries
  - Other Libraries
  - Depletion chains

## OpenMC

[Home](#) [Documentation](#) [Data](#) [Discussion Forum](#)

### Official Data Libraries

The data libraries listed here have been produced by the OpenMC development team and are as complete as possible. HDF5 files are created by first processing source ENDF files into ACE files and then using the [openmc\\_data](#) Python module to convert ACE data into HDF5. The entire process is automated by the [IncidentNeutronFromNJOY](#) method which calls [NJOY](#) under the hood to produce ACE files.

#### ENDF/B-VII.1

This library includes incident neutron, photoatomic, thermal scattering, and windowed multipole data. All ACE files were produced using NJOY 2016.44. Incident neutron data is available at six temperatures: 250 K, 293.6 K, 600 K, 900 K, 1200 K, and 2500 K. Note that elastic scattering cross sections are also available at 0 K and can be used for modeling resonance upscattering in heavy nuclides. Thermal scattering data is available at the tabulated temperatures from the source ENDF files. Windowed multipole data can be used to evaluate temperature-dependent cross sections at run-time.

Download: [[tar.xz](#)]

#### ENDF/B-VIII.0

This library includes incident neutron, photoatomic, atomic relaxation, and thermal scattering data from ENDF/B-VIII.0. All ACE files were produced using NJOY 2016.59. Incident neutron data is available at six temperatures: 250 K, 293.6 K, 600 K, 900 K, 1200 K, and 2500 K. Thermal scattering data is available at the tabulated temperatures from the source ENDF files.

Downloads: [[tar.xz](#)]

#### JEFF 3.3

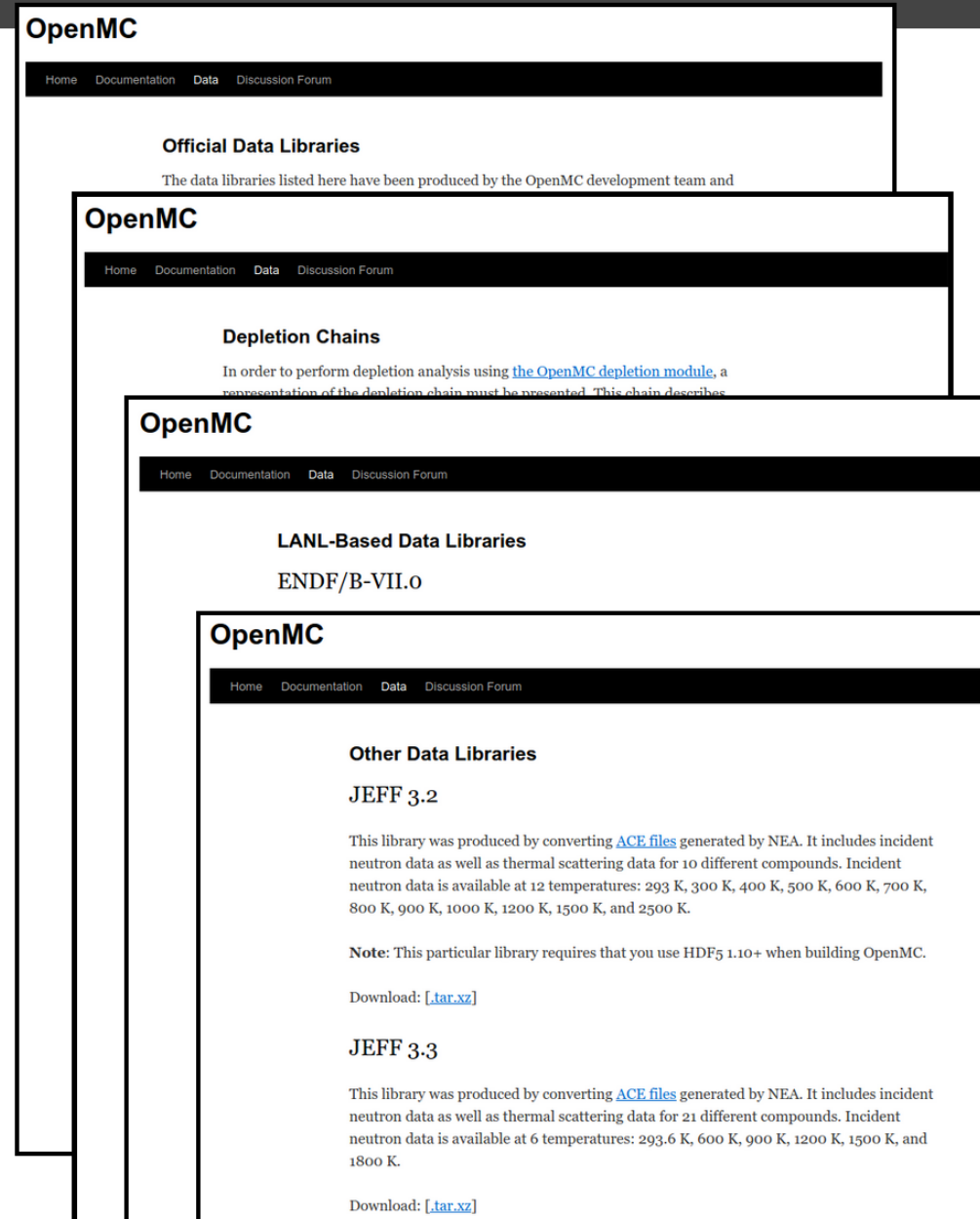
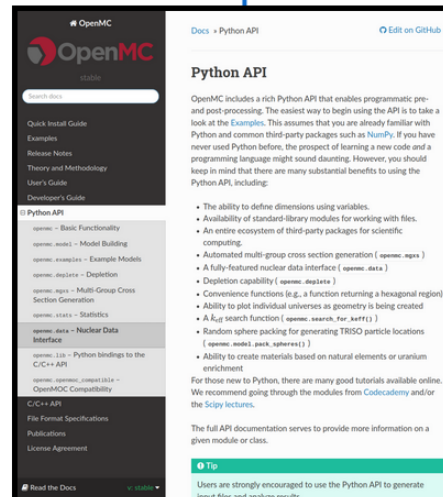
This library includes incident neutron and thermal scattering data from JEFF 3.3. All ACE files were produced using NJOY 2016.59. The photoatomic and atomic relaxation data from ENDF/B-VIII.0 has also been included because JEFF 3.3 does not release its own photoatomic or atomic relaxation sublibraries. Incident neutron data is available at six temperatures: 250 K, 293.6 K, 600 K, 900 K, 1200 K, and 2500 K. Thermal scattering data is available at the tabulated temperatures from the source ENDF files.

Note that the C13 and O17 cannot be processed by NJOY 2016.59 due to a bug in the evaluations. These nuclides have been replaced by data from TENDL-2019 (the original evaluations from JEFF 3.3 were taken from TENDL-2015).

Download: [[tar.xz](#)]

# OpenMC.org

- <https://openmc.org> includes a data section with downloads
  - Official Libraries
  - LANL-Based Data Libraries
  - Other Libraries
  - Depletion chains
- OpenMC documentation pages includes information on the nuclear data interface  
<https://docs.openmc.org>



# Obtaining h5 cross sections - OpenMC.org

- Downloads can be scripted using wget
- The h5 files can be used in simulations once the OPENMC\_CROSS\_SECTION environmental variable has been set

```
wget -O nndc-b7.1.tar.xz https://anl.box.com/shared/static/9igk353zpy8fn9ttvtrqgzvwlvttejoz6.xz
>> downloading compressed file ...

mkdir nndc-b7.1-hdf5

tar -xf nndc-b7.1.tar.xz -C nndc-b7.1-hdf5
>> extracting compressed file ...

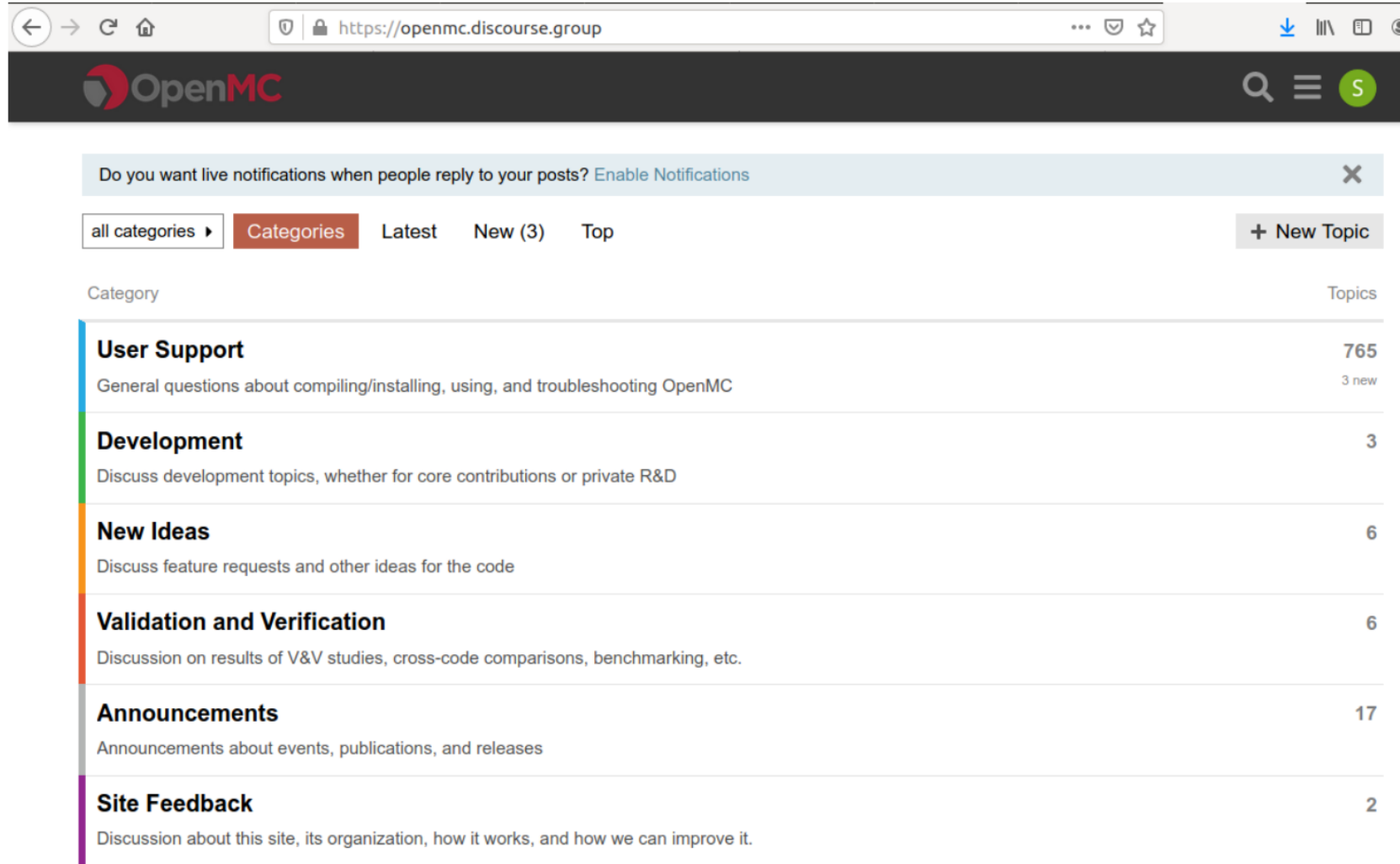
ll nndc-b7.1-hdf5
>>cross_sections.xml
>>neutron/
>>photon/

ll nndc-b7.1-hdf5/neutron
>>Ac225.h5
>>Ac226.h5
>>Ac227.h5
>>Ag107.h5
>>Ag109.h5
>>...

export OPENMC_CROSS_SECTIONS=nndc-b7.1-hdf5/cross_sections.xml
```



# Obtaining h5 cross sections - OpenMC.org



The screenshot shows the OpenMC.org website interface. At the top, there is a navigation bar with the OpenMC logo, a search icon, and a user profile icon. Below the navigation bar, there is a notification banner asking if the user wants live notifications. The main content area features a list of categories with their respective topic counts and descriptions. The categories are: User Support (765 topics), Development (3 topics), New Ideas (6 topics), Validation and Verification (6 topics), Announcements (17 topics), and Site Feedback (2 topics).

Category	Topics
<b>User Support</b> General questions about compiling/installing, using, and troubleshooting OpenMC	765 3 new
<b>Development</b> Discuss development topics, whether for core contributions or private R&D	3
<b>New Ideas</b> Discuss feature requests and other ideas for the code	6
<b>Validation and Verification</b> Discussion on results of V&V studies, cross-code comparisons, benchmarking, etc.	6
<b>Announcements</b> Announcements about events, publications, and releases	17
<b>Site Feedback</b> Discussion about this site, its organization, how it works, and how we can improve it.	2

# OpenMC Docker image production

- OpenMC automatically produces a new docker image with every code merge
- Github Actions is used to build the docker image and upload it to Dockerhub



A screenshot of the GitHub Actions interface for the repository `openmc-dev/openmc`. The `dockerhub-publish-develop` workflow is selected, showing a list of 51 workflow runs. Each run is a green checkmark indicating a successful merge pull request. The runs are triggered by merge pull requests from various contributors like `gridley/ref...`, `openmc-d...`, `gridley/pr...`, `shikhar41...`, `ltramm/se...`, `gridley/ad...`, `paulroma...`, and `paulroma...`. The interface includes a search bar, filters for event, status, branch, and actor, and a table of workflow runs with columns for event, status, branch, and actor.

<https://github.com/openmc-dev/openmc/actions>



A screenshot of the Docker Hub page for the `openmc/openmc` image. The page shows the image name, the owner `openmc`, and the update date (3 days ago). The description states: "The OpenMC particle transport code (https://github.com/openmc-dev/openmc) on Ubuntu 18.04." Below the description, there is a section for the Docker Pull Command, which is `docker pull openmc/openmc`. The page also includes a section for the owner's profile, showing the name `openmc`. The interface includes a search bar, navigation links for Explore, Pricing, Sign In, and Sign Up, and a table of workflow runs.

<https://hub.docker.com/l/openmc/openmc>

# OpenMC Docker image production

- OpenMC automatically produces a new docker image with every code merge
- Github Actions is used to build the docker image and upload it to Dockerhub



A screenshot of the GitHub Actions interface for the repository openmc-dev/openmc. The 'Actions' tab is selected, showing a workflow named 'dockerhub-publish-develop'. A table lists 51 workflow runs, with the most recent ones being merge pull requests. Each entry shows the commit hash, the actor, and the time since the run.

Event	Status	Branch	Actor
Merge pull request #1784 from gridley/ref...	develo	develop	3 days ago
Merge pull request #1793 from openmc-d...	develo	develop	3 days ago
Merge pull request #1792 from gridley/pr...	develo	develop	4 days ago
Merge pull request #1759 from shikhar41...	develo	develop	5 days ago
Merge pull request #1764 from [tramm/se...	develo	develop	7 days ago
Merge pull request #1782 from gridley/ad...	develo	develop	7 days ago
Merge pull request #1785 from paulroma...	develo	develop	7 days ago
Merge pull request #1776 from paulroma...	develo	develop	10 days ago

<https://github.com/openmc-dev/openmc/actions>



A screenshot of the Docker Hub page for the image openmc/openmc. The page shows the repository name, the owner 'openmc', and the update time 'Updated 3 days ago'. It also displays the Docker pull command and the owner's profile link.

**openmc/openmc** ☆  
By openmc • Updated 3 days ago  
The OpenMC particle transport code (https://github.com/openmc-dev/openmc) on Ubuntu 18.04.  
Container

**Docker Pull Command**

```
docker pull openmc/openmc
```

**Owner**  
profile openmc

<https://hub.docker.com/r/openmc/openmc>

# Data storage repo - processing

Contains repository for the openmc-data-downloader Python package.

Contains separate repositories for each library release

Repositories can be templated

Open source, permissively licensed (MIT)  
<https://github.com/openmc-data-storage>

The screenshot shows the GitHub interface for the repository 'openmc-data-storage'. The page title is 'openmc-data-storage' and it features a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are statistics for Repositories (5), Packages, People (2), Teams, Projects, and Settings. A search bar for finding repositories is present, along with filters for Type, Language, and Sort. The main content area displays a list of repositories:

- openmc\_data\_downloader**: Download cross section h5 files for use in OpenMC. Language: Python. License: MIT. Updated 2 hours ago.
- JEFF-3.2**: Nuclear data from the JEFF-3.3 library process in the OpenMC h5 file format. License: MIT. Updated 11 hours ago.
- FENDL-3.1d**: Nuclear data from the FENDL-3.1d library processed in the OpenMC h5 file format. License: MIT. Updated yesterday.
- ENDF-B-VII.1-NNDC**: Nuclear data from the ENDF-B-VII.1 NNDC library processed in the OpenMC h5 file format. License: MIT. Updated 4 days ago.
- TENDL-2019**: Nuclear data from the TENDL 2019 library processed in the OpenMC h5 file format. License: MIT. Updated 4 days ago.

Red arrows from the text on the left point to the corresponding repository entries in the screenshot.

# Data storage repo - environment

4 Lines



Use of the openmc/openmc-dev image  
this contains NJOY as well

Addition of the openmc-data  
repo to the base image

Copying test suite

See environmental variable

```
FROM openmc/openmc:develop
RUN git clone https://github.com/openmc-dev/data.git
COPY tests tests/
ENV OPENMC_CROSS_SECTIONS=/share/cross_sections.xml
```

# Data storage repo - processing

4 Lines



+

40 Lines



+



=



GitHub actions (CI) build and run the new docker image



The new docker image is then run externally with a shared volume mount



GitHub action (CI) then commits the new h5 files from the share folder to the repository

```
- name: Create custom readme
  run: |
    python readme_writer.py
    cat README.md
- name: Build and test with Docker
  run: |
    docker build --tag nuc_data_maker .
    rm -rf h5_files
    docker run -v /home/runner/work/${{ github.event.repos
- name: Commit files
  run: |
    git config --local user.email "action@github.com"
    git config --local user.name "GitHub Action"
    git add h5_files/*.h5
    git add h5_files/cross_sections.xml
    git add README.md
    git commit -m "Added readme with repo name (bot commit)"
- name: Push changes
  uses: ad-m/github-push-action@master
  with:
    github_token: ${{ secrets.GITHUB_TOKEN }}
    branch: ${{ github.ref }}
```

# Data storage repo - testing

4 Lines



+

40 Lines



+

39 Lines



=



```
import xml.etree.ElementTree as ET

import openmc

os.environ["OPENMC_CROSS_SECTIONS"] = '/share

class TestH5FilesInSimulations(unittest.TestCase)

    def test_simulation_runs_with_isotopes(self):
        tree = ET.parse('/share/h5_files/cross
        root = tree.getroot()

        for elem in root:
            if elem.attrib['type'] == 'neutro
                isotope_name = elem.attrib['m

                sett = openmc.Settings()
                sett.batches = 100
                sett.inactive = 0
                sett.particles = 500
                sett.run_mode = 'fixed source'
```

Each h5 cross section file is then used in an OpenMC simulation

Testing the h5 file produced has converted successfully and does not cause errors during usage

```
build
succeeded 1 hour ago in 8m 14s

Build and test with Docker 6m 16s

1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578

| The OpenMC Monte Carlo Code
Copyright | 2011-2021 MIT and OpenMC contributors
License | https://docs.openmc.org/en/latest/license.html
Version | 0.12.1-dev
Git SHA1 | c4e8881d07e79042d5a69f9cae6e373963cfd1
Date/Time | 2021-03-17 10:16:02
MPI Processes | 1
OpenMP Threads | 2

Reading settings XML file...
Reading cross sections XML file...
Reading materials XML file...
Reading geometry XML file...
Reading H5 from /share/h5_files/neutron/Hed4.h5
Minimum neutron data temperature: 300.0 K
Maximum neutron data temperature: 300.0 K
Preparing distributed cell instances...
Writing summary.h5 file...
Maximum neutron transport energy: 60000000.0 eV for Hed4

===== FIXED SOURCE TRANSPORT SIMULATION =====
Simulating batch 1
Simulating batch 2
```

# Data storage repo - storage

4 Lines



+

40 Lines

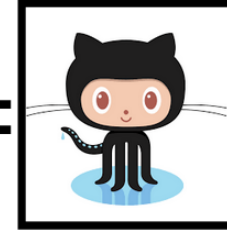


+

39 Lines



=



Gigabytes of h5 files each with their own URL endpoint

[https://github.com/openmc-data-storage/FENDL-3.1d/blob/main/h5\\_files/neutron/Ag107.h5?raw=true](https://github.com/openmc-data-storage/FENDL-3.1d/blob/main/h5_files/neutron/Ag107.h5?raw=true)

actions-user Added readme with repo name (bot commit) 82af7e9 1 hour ago History		
..		
Ag107.h5	Added readme with repo name (bot commit)	1 hour ago
Ag109.h5	Added readme with repo name (bot commit)	1 hour ago
Al27.h5	Added readme with repo name (bot commit)	1 hour ago
Ar36.h5	Added readme with repo name (bot commit)	1 hour ago
Ar38.h5	Added readme with repo name (bot commit)	1 hour ago
Ar40.h5	Added readme with repo name (bot commit)	1 hour ago
Au197.h5	Added readme with repo name (bot commit)	1 hour ago
B10.h5	Added readme with repo name (bot commit)	1 hour ago
B11.h5	Added readme with repo name (bot commit)	1 hour ago
Ba130.h5	Added readme with repo name (bot commit)	1 hour ago
Ba132.h5	Added readme with repo name (bot commit)	1 hour ago
Ba134.h5	Added readme with repo name (bot commit)	1 hour ago
Ba135.h5	Added readme with repo name (bot commit)	1 hour ago
Ba136.h5	Added readme with repo name (bot commit)	1 hour ago
Ba137.h5	Added readme with repo name (bot commit)	1 hour ago
Ba138.h5	Added readme with repo name (bot commit)	1 hour ago
Be9.h5	Added readme with repo name (bot commit)	1 hour ago
Bi209.h5	Added readme with repo name (bot commit)	1 hour ago
Br79.h5	Added readme with repo name (bot commit)	1 hour ago
Br81.h5	Added readme with repo name (bot commit)	1 hour ago



# Data storage repo - openmc-data-downloader

Easy to install Python package

```
pip install openmc_data_downloader
```

Knowledge of all the URLs of all the h5 files available.

```
{
  'element': 'Zn',
  'isotope': 'Zn66',
  'library': 'FENDL-3.1d',
  'local_file': 'FENDL-3.1d_Zn66.h5',
  'remote_file': 'Zn66.h5',
  'url': 'https://github.com/openmc-data-storage/FEN
  },
  ...
}
```

Internal data based that gets filtered  
by library / isotopes etc



Translation from user instructions to a  
list of h5 files to download

A screenshot of a GitHub repository page for 'openmc-data-storage / openmc\_data\_downloader'. The page shows the repository name, navigation tabs (Code, Issues, Pull requests, Actions, Projects, Wiki, Security), and a commit history table. The commit history table lists recent commits with their messages and timestamps. Below the commit history, there is a section for the README.md file, which includes CI status indicators and the title 'OpenMC data downloader'. The README text describes the package as a Python tool for downloading OpenMC data in h5 format.

Commit	Message	Time
Shimwell	bump version	2 hours ago
	corrected docker tag name	2 days ago
	using - and . in library names instead of _	17 hours ago
	using - and . in library names instead of _	17 hours ago
	added openmc files xml h5	2 days ago
	using python setup instead of pip	2 days ago
	Initial commit	4 days ago
	using - and . in library names instead of _	17 hours ago
	added ability to read materials.xml file	3 days ago
	added ci for tests and enviroment	2 days ago
	bump version	2 hours ago

[https://github.com/openmc-data-storage/openmc\\_data\\_downloader](https://github.com/openmc-data-storage/openmc_data_downloader)



# Data storage repo - openmc-data-downloader

```
openmc_data_downloader -l FENDL-3.1d -i Li6
```

Downloads one isotope

```
openmc_data_downloader -l TENDL-2019 -i Li6 Li7
```

Downloads two isotopes

```
openmc_data_downloader -l TENDL-2019 -e Li
```

Downloads one element

```
openmc_data_downloader -l TENDL-2019 -e Li Si Na
```

Downloads two elements

```
openmc_data_downloader -l ENDFB-7.1-NNDC -i Be9 -d my_h5_files
```

Downloads to a directory

```
openmc_data_downloader -l TENDL-2019 -e Li Si Na -i Fe56 U235
```

Downloads a combination of elements and isotopes

```
openmc_data_downloader -l TENDL-2019 -m materials.xml
```

Downloads all isotopes in a materials.xml file

```
openmc_data_downloader -l FENDL-3.1d TENDL-2019 -i Li6 U235
```

Downloads isotopes with library preference

# Just in time library generator

```
import openmc
import openmc_data_downloader as odd

mat1 = openmc.Material(1, "breeder_mat")
mat1.add_element('Pb', 84.2, percent_type='ao')
mat1.add_element('Li', 15.8, percent_type='ao')
mat1.set_density('g/cm3', 11.3)
mats = openmc.Materials([mat1])

odd.just_in_time_library_generator(
    libraries='FENDL-3.1d',
    materials=mat1
)
```

← Extra import

← Material

← Download h5 cross section files for the materials, accepts lists or individual items

Resulting download of h5 files and setting of OPENMC\_CROSS\_SECTIONS environmental variable

Re-running the same Python script and h5 downloading can be skipped as they already exist

```
(base) jshim@jshim-desktop:~$ python iaea_example.py
Searching libraries with the following priority {'FENDL-3.1d': 1}
Isotopes found matching library requirements 180
Isotopes found matching isotope requirements 18
Isotopes found matching all requirements 6
Downloading FENDL-3.1d_Li6.h5...
Downloading FENDL-3.1d_Li7.h5...
Downloading FENDL-3.1d_Pb204.h5...
Downloading FENDL-3.1d_Pb206.h5...
Downloading FENDL-3.1d_Pb207.h5...
Downloading FENDL-3.1d_Pb208.h5...
/home/jshim/cross_sections.xml written
setting OPENMC_CROSS_SECTIONS /home/jshim/cross_sections.xml
```

```
(base) jshim@jshim-desktop:~$ python iaea_example.py
Searching libraries with the following priority {'FENDL-3.1d': 1}
Isotopes found matching library requirements 180
Isotopes found matching isotope requirements 18
Isotopes found matching all requirements 6
Skipping FENDL-3.1d_Li6.h5, already downloaded
Skipping FENDL-3.1d_Li7.h5, already downloaded
Skipping FENDL-3.1d_Pb204.h5, already downloaded
Skipping FENDL-3.1d_Pb206.h5, already downloaded
Skipping FENDL-3.1d_Pb207.h5, already downloaded
Skipping FENDL-3.1d_Pb208.h5, already downloaded
/home/jshim/cross_sections.xml written
setting OPENMC_CROSS_SECTIONS /home/jshim/cross_sections.xml
```



# Scale using repository templates



Less time spent making a repo for new nuclear data

~ 1 min human time

Architecture +  
customization = new  
nuclear data repo

Live demo

# Building blocks

OpenMC Python API

↓ IncidentNeutron, from\_endf and from\_ace  
+ URL endpoints for ACE and ENDF libraries

Openmc-dev data repository

↓ convert\*.py and generate\*.py  
+ Run commands and containerized environment

Automated processing and  
testing via CI

↓ docker build, docker run and pytest  
+ URL endpoints for h5 files and libraries

Downloading of reproducible  
collections of isotopes

# Summary

There are several methods of obtaining the h5 cross section files needed to run OpenMC simulations.

Each method has a specific use case.

- OpenMC.org - GUI, script, entire libraries, h5 data
- OpenMC-dev data repo - scripts for processing libraries
- openmc-data-downloader - reproducible mixed micro libraries

Future work

- Ability to specify particle type to allow photons
- Ability to specify temperature