# How to convert dicom to BIDS for RESSTORE

**USAGE:** This document describes how to convert dicom files to the BIDS format for the study RESSTORE.

# PREREQUISITES:

- verify that you use python3 or else replace all the commands with python3 by python
- install dcm2niix
- download the library dcm2bids and the two files 'add data.py' and 'config.json'

## **DISCLAIMERS:**

- The terms visit and session refer to the same thing and should have the same numbers.
- The following instructions allow you to start from scratch. To update the database when a new patient or visit is added you just have to do the STEP 4.

**STEP 0:** To verify that the config.json files match with the description of your DICOM files and modify them if necessary, you need to:

**STEP 1:** Create a working directory containing a combination of Python and shell commands, including 'add\_data.py' script and 'config.json' file and a folder BIDS\_sourcedata which will contain all the dicom files. The directory should look as described below:

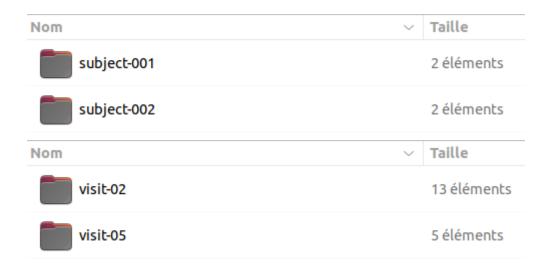
BIDS_sourcedata	3 éléments
add_data.py	4,0 ko
config.json	4,8 ko

**STEP 2:** Organize your BIDS\_sourcedata folder as described below. Customize the directory structure so that each subject has a folder named "<u>subject-XXX</u>" containing subfolders for each visit entitled "<u>visit-XX</u>", and ensure **NO** .par or .rec files are included (if there is one, delete the parrec directory).

Important: name the directories as shown.

- Create folders for each subject (subject-XXX).
- Inside each subject folder, create subfolders for each visit (visit-XX).

For example:

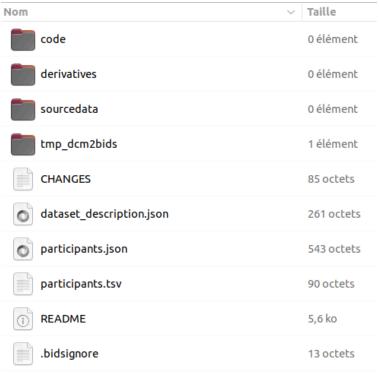


**STEP 3:** Create an output directory and organize it by using the following script:

mkdir OUTPUT\_DIR

dcm2bids\_scaffold -o OUTPUT\_DIR

The OUTPUT\_DIR should look like this:



**STEP 4:** Run the code with the following script:

python3 add\_data.py -s /path/to/BIDS\_sourcedata/ -o /path/to/OUTPUT\_DIR -c /path/to/config.json

This step is a bit long. It's important to check the pairing at the end. If the output message indicates 'no pairing' for a useful file you should verify your config.json file. For instance:

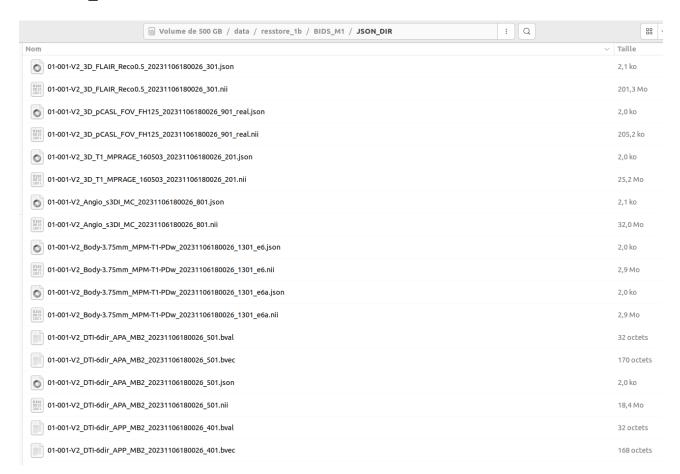
The dicom file corresponding to the T1 for the subject 002, visit 05 has successfully been converted to BIDS format. However, the "visit-05\_Cor" has not. If it's not a useful file, ignore the message. If it is, try the additional step.

ADDITIONAL STEP IF SOMETHING DOESN'T WORK: You probably need to edit the config.json file. To do so you can do a first conversion of your problematic dicom with this command:

```
mkdir JSON_DIR
```

```
dcm2niix -o /path/to/JSON_DIR
/path/to/the/folder/of/one/specific/visit/for/one/specific/subject
```

Be careful this command only deals with one patient at each time. Once this step is done, the JSON DIR looks like this:



First verify that your file has been successfully converted to a JSON format.

Then, Ensure the file aligns with the specifications in the config.json file. This includes:

- SeriesDescription: Confirm that the description matches.
- Name: Verify that the name is correct.
- Suffix: Check that the suffix is appropriate and accurate.

**Important**: For the DWI sequence, make sure to add the phase encoding direction in the JSON config file (with special attention on Philips NMR machines).

- For PA direction the phase encoding is "j"
- For AP direction the phase encoding is "j-"

Ex:

```
"datatype": "dwi",
    "suffix": "dwi",
    "custom_entities": "acq-abcd2_dir-PA",
    "criteria": {
        "SeriesDescription": "DTI 2_2mm"
    },
    "sidecar_changes": {
        "PhaseEnPhasecodingDirection": "j"
    }
},
```

**HOW TO SEE ALL THE JSON FILES**: Create an intermediate JSON directory and run the code with the following script:

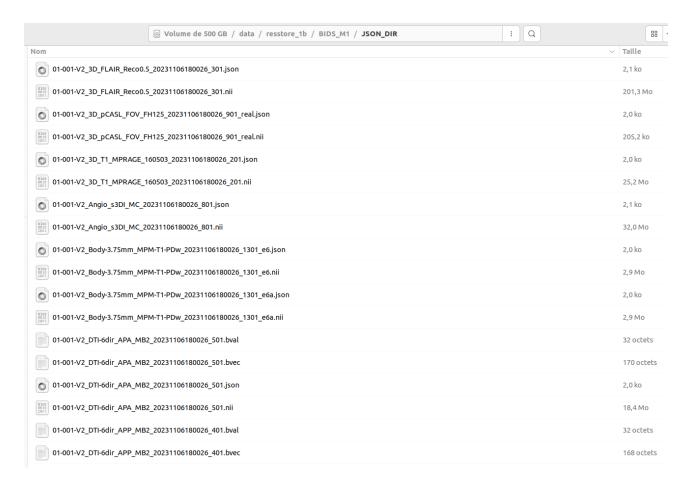
```
mkdir JSON_DIR
```

```
dcm2niix -o dossier_stockage_json_intermediaire chemin_d'acces_fichier_patient
```

# for example:

dcm2niix -o '/Path/toJSON\_DIR' '/Path/to/BIDS\_M1/01-001-V2'

The JSON directory will look like this:



### STUDIES FOR EACH PATIENT:

With the provided config.json file, for each patient after conversion, the following files will be organized according to BIDS format:

- 'anat' for anatomical data
  - T1: This may include 3D T1, sT1w, MPRAGE.
  - FLAIR: 3D FLAIR.
- 'dwi' for diffusion data and 'fmap' for field map data

# Two Diffusion protocols are used:

- 1- Hermes (single shell) with a 'dwi\_h' directory including DTI 60dir file and a 'fmap\_h' directory including DTI 6dir APA, DTI 6dir APP.
- 2- **ABCD** (multishell) with a 'dwi' directory including DTI 1 and DTI 2 files and a 'fmap' directory including DTI Fieldmap A and DTI Fieldmap P. In case a patient had both studies, all the corresponding files will be present.
- 'func' for functional data: This is an optional file located within the func folder, which may include task-related fMRI (e.g. Left Hand or Paretic Hand tasks) or resting-state fMRI.