

# Convert DICOM to BIDS

## Overview

### What You Need

- Scripts

### Make the configuration file

- Using the Dcm2Bids\_helper

### Steps to convert DICOMS to BIDS

### Manually Create Metadata Files

### Check the BIDS Conversion

## Overview

DICOMS are converted into Niftis, which are renamed and put into BIDS structure using [cbedetti's dcm2Bids](#) package. The dcm2Bids package converts DICOM files to Nifti files using the [rordenlab's dcm2niix](#) package, then renames and relocates them as per [BIDS](#) specifications.

**Instructions on this page are to run the scripts on a high performance cluster running SLURM using a Singularity container of the dcm2Bids package. If you want to run the dcm2bids helper or the dcm2bids conversion on a local machine, see [cbedetti's original dcm2Bids package as the scripts described below will not work for you.](#)**

## What You Need

- **Dcm2Bids singularity container** (exists at `/projects/sanlab/shared/containers/Dcm2Bids-master.simg`)
  - If you need a copy of the container to exist elsewhere, see [Containers: Docker & Singularity](#) (and, as always, read the [documentation](#))
  - Note that if you create a new container, you either need to give it the same name ( `Dcm2Bids-master.simg` ) or change the image name in the `config_dcm2bids_batch.py` script.

The following files should be in a single directory (e.g. `REV_scripts/org/dcm2bids` ). You can pull them from the [git repo](#). After you clone the repo, copy the "dcm2bids" directory into your own "StudyName\_scripts/org" folder, and remove the hidden `.git` directory within that directory using the following code:

```
cd StudyName_scripts/org/dcm2bids
rm -rf .git
```

## Scripts

- `dcm2bids_helper.py`
- `config_dcm2bids_helper.py` - *change the variables and paths as appropriate for your study*
- `dcm2bids_batch.py`
- `config_dcm2bids_batch.py` - *change the variables and paths as appropriate for your study*
- `subject_list.txt` - *populate this text file with a list of subjects you want to convert.*
- `study_config.json` - the study config file (instructions on how to make it for your study are [here](#))

## Make the configuration file

### Using the Dcm2Bids\_helper

If you need the metadata to populate the config file, use the dcm2bids helper, which is built into the dcm2bids container.

1. Create the `subject_list.txt`
  - An easy way to do this is to `cd` into your DICOM directory and use the command `ls >> subject_list.txt` , which will

create a text file of that name and populate it with all the directories/files in your working directory. You can then move that subject\_list.txt file to the directory from which you will be running your code and add the subject ID and time-point columns.

- The subject\_list should be formatted such that each row consists of: the subject directory names (that contains the dicoms), desired subject ID, and data collection wave number. Each field is comma separated, all without spaces, e.g.:

```
sub01_20150909,REV001,wave1
sub01_20150909,REV001,wave2
sub02_20150909,REV001,wave1
```

2. Change the variables and/or paths in the config\_dcm2bids\_helper.py script for your study

3. Log into Talapas

```
ssh -X username@Talapas-ln1.uoregon.edu
```

4. cd to the directory that has your code in it

5. Load the python3 module

```
module load python3
```

6. Run the dcm2bids\_helper.py script

```
python3 dcm2bids_helper.py
```

7. cd to the folder created by the helper (should be in the top level of your study directory), e.g.

```
cd /projects/sanlab/shared/REV/tmp_dcm2bids/helper
ls

001_REV001_20150406_AAHSout_20150406145550.nii.gz
002_REV001_20150406_AAHSout_20150406145550a.json
.
.
.
017_REV001_20150406_React2_mb3_g2_2mm_te27_20150406145550.json
017_REV001_20150406_React2_mb3_g2_2mm_te27_20150406145550.nii.gz
```

8. View the json files and use that info to edit the config file so it works for your study. Instructions in the readme of the [dcm2Bids](#) repo.

## Steps to convert DICOMS to BIDS

1. Create a subject list where each row has the input: `directoryName,subjectID,waveNumber`
2. Edit the config file such that it works for your study.
  - a. See the [dcm2Bids](#) repository for documentation and instructions. for real, you'll need to read the instructions

3. Change the variables and/or paths in the `config_dcm2bids_batch.py` script for your study
4. Log into Talapas

```
ssh -X username@Talapas-ln1.uoregon.edu
```

5. `cd` to the directory that has your code in it
6. Load the python3 module

```
module load python3
```

7. Run the batch script

```
python3 dcm2bids_batch.py
```

8. Check the niftis, output logs, and error logs.

## Manually Create Metadata Files

As per: [http://bids.neuroimaging.io/bids\\_spec1.0.0-rc2.pdf](http://bids.neuroimaging.io/bids_spec1.0.0-rc2.pdf)

Place these files in the top level bids directory.

- `dataset_description.json`
- `phasediff.json`
- A JSON for each functional task with `TaskName` and `RepetitionTime`
- README (optional, but strongly recommended)
- CHANGES (optional, but strongly recommended)

## Check the BIDS Conversion

BIDS validator: <http://incf.github.io/bids-validator>