

Tile Number of Surface Reflectance.

Tile Size: 6° x 6°

Naming rule: H8_{datetime}_h{xx}v{xx}_LSR.nc

e.g. H8_202301010000_h08v04_LSR.nc

Read **example**. (I would recommend using xarray to read the file.)

(<https://docs.xarray.dev/en/latest/getting-started-guide/installing.html>)



































```
#####
```

```
import xarray as xr
```

```
path = '/h08v04/2023/01/01/H8_202301010000_h08v04_LSR.nc'
```

```
ds = xr.open_dataset(path)
```

```
#####
```

► Dimensions:	(lat_1km: 600, lon_1km: 600, lat_2km: 300, lon_2km: 300)			
▼ Coordinates:				
lat_1km	(lat_1km)	float32	35.99 35.99 35.97 ... 30.01 30.0	 
lon_1km	(lon_1km)	float32	133.0 133.0 133.0 ... 139.0 139.0	 
lat_2km	(lat_2km)	float32	35.99 35.97 35.95 ... 30.03 30.01	 
lon_2km	(lon_2km)	float32	133.0 133.0 133.1 ... 139.0 139.0	 
▼ Data variables:				
Band01	(lat_1km, lon_1km)	int16	...	 
[360000 values with dtype=int16]				
Band02	(lat_1km, lon_1km)	int16	...	 
Band03	(lat_1km, lon_1km)	int16	...	 
Band04	(lat_1km, lon_1km)	int16	...	 
Band05	(lat_2km, lon_2km)	int16	...	 
Band06	(lat_2km, lon_2km)	int16	...	 
Solar Zenith An...	(lat_2km, lon_2km)	float32	...	 
Solar Azimuth A...	(lat_2km, lon_2km)	float32	...	 
View Zenith An...	(lat_2km, lon_2km)	float32	...	 
View Azimuth A...	(lat_2km, lon_2km)	float32	...	 
Cloud Mask(2km)	(lat_2km, lon_2km)	float32	...	 
▼ Indexes:				
lat_1km	PandasIndex			
lon_1km	PandasIndex			
lat_2km	PandasIndex			
lon_2km	PandasIndex			

Spatial resolution:

0.01deg for Band01~04

0.02deg for Band05/06 , Cloud mask, angular data.

SR data information

Nan value: -32768

Valid_Range : 0 ~ 10000

Slope: 0.0001

Code example

```
import numpy as np
import matplotlib.pyplot as plt

AHI_B4 = np.array(ds['Band04'])
AHI_B4 = np.where(AHI_B4 == -32768, np.nan, AHI_B4 / 10000)
```

```
plt.imshow(AHI_B4)
plt.colorbar()
```

<matplotlib.colorbar.Colorbar at 0x7f796a891e80>

