

# Finding Extracted Features data for a known volume ID

The filepath to sync [Extracted Features](#) files through RSync follows a [pairtree format](#), keeping the institutional shortcode intact (e.g. mpd, uc2).

## Converting ID to RSync URL (Python with HTRC Feature Reader library)

If you are the HTRC Feature Reader library, there is a convenience function in `htrc_features.utils.id_to_rsync(htid)`:

```
>> from htrc_features import utils
>> utils.id_to_rsync('miun.adx6300.0001.001')
'miun/pairtree_root/ad/x6/30/0,/00/01/,0/01/adx6300,0001,001/miun.adx6300,0001,001.json.bz2'
```

## Converting ID to RSync URL (Python)

This example is a simplified part of a longer notebook, which further describes how to collect and download large lists of volumes: [ID to EF Rsync Link](#). [ipynb](#).

If you don't have it, you may have to install the pairtree library with: `pip install pairtree` (Python 2.x only).

```
import os
from pairtree import id2path, id_encode
def id_to_rsync(htid):
    '''
        Take an HTRC id and convert it to an Rsync location for syncing Extracted Features
    '''
    libid, volid = htid.split('.', 1)
    volid_clean = id_encode(volid)
    filename = '.'.join([libid, volid_clean, kind, 'json.bz2'])
    path = '/'.join([kind, libid, 'pairtree_root', id2path(volid).replace('\', '/'), volid_clean, filename])
    return path
```

Example:

```
id_to_rsync('miun.adx6300.0001.001')
'miun/pairtree_root/ad/x6/30/0,/00/01/,0/01/adx6300,0001,001/miun.adx6300,0001,001.json.bz2'
```

The Extracted Features for this volume can be downloaded using RSync:

```
rsync -azv data.analytics.hathitrust.org::features/{{URL}} .
```