

# Accessing Surface Data

---

If you can't find the data file in the location for the current data

## Valpo Local Data Access

Local Access Current Data: `/data/ldmdata/surface/sao`

```
from metpy.io import metar

data = '/data/ldmdata/surface/sao/20220114_sao.wmo'

df = metar.parse_metar_file(data, year=2021, month=1)
```

## Valpo Remote Data Access

Remote Access Current Data (Valpo: [http://bergeron.valpo.edu/current\\_surface\\_data/](http://bergeron.valpo.edu/current_surface_data/)):

```
from io import StringIO
from urllib.request import urlopen

from metpy.io import metar

data = StringIO(urlopen('http://bergeron.valpo.edu/current_surface_data/'
                        '20201221_sao.wmo').read().decode('utf-8',
                    'backslashreplace'))

df = metar.parse_metar_file(data, year=2020, month=12)
```

## Iowa State Archive Remote Data Access

Remote Access Archived Data (Iowa State Archive: 1950-current):

```
from io import StringIO
from urllib.request import urlopen

from metpy.io import metar
import pandas as pd

# Set a date and time for the observations you want
date = datetime(1993, 3, 13, 12)

# Set a time window to look for observations
dt = timedelta(minutes=15)
```

```
# Use datetime and time window to set final start/end times
sdate = date - dt
edate = date + dt

# Use Pandas remote csv reading capability to grab desired data
# Make Missing values ('\M\') into NaNs
# Replace Trace precip with very small float value

data = pd.read_csv('http://mesonet.agron.iastate.edu/'
                   'cgi-bin/request/asos.py?'
                   data=all&tz=Etc/UTC&format=comma&
                   f'latlon=yes&elev=yes&year1={sdate.year}&month1=
{sdate.month}&'
                   f'day1={sdate.day}&hour1={sdate.hour}&minute1=
{sdate.minute}&'
                   f'year2={edate.year}&month2={edate.month}&day2=
{edate.day}&'
                   f'hour2={edate.hour}&minute2={edate.minute}',
skiprows=5,
                   na_values=['M'],
                   low_memory=False).replace('T',
0.00001).groupby('station').tail(1)

# Parse METAR observations using MetPy
df = metar.parse_metar_file(StringIO('\n'.join(val for val in
data.metar)),
                           year=date.year, month=date.month)
```