Geographic data information is used by several programs to display map overlays.

The geographic data files include:

File Name	Contents
coord_<user_id>.dat	Information to define the subset of the national HRAP grid
county.dat	County boundaries
fg_basin.dat	Forecast Group boundaries
flights.dat	Flight lines
forecastpt.dat	Forecast points
map_basin.dat	Basin boundaries
rfc_boundary.dat	Forecast area boundary
river.dat	River boundaries
state.dat	State boundaries
town.dat	Town locations
town_zoom.dat	Town locations for zooming in

The files coord_<user_id>.dat and rfc_boundary.bin files must exist to run the Operational Forecast System Interactive Forecast Program (IFP).

The steps to create the geographic data files are:

1. **Create a directory structure for geo data for <user_id>**:

   ```
   mkdir $(geo_data)/<user_id>
   mkdir $(geo_data)/<user_id>/ascii
   mkdir $(geo_data)/<user_id>/binary
   ```

2. **Make sure that the necessary apps_defaults are set to <user_id>**.

   At a minimum the apps_default ifp_rfc should be <user_id>.

3. **Create the file coord_<user_id>.dat in the ascii directory**.

   The values in the coord_<user_id>.dat file are:

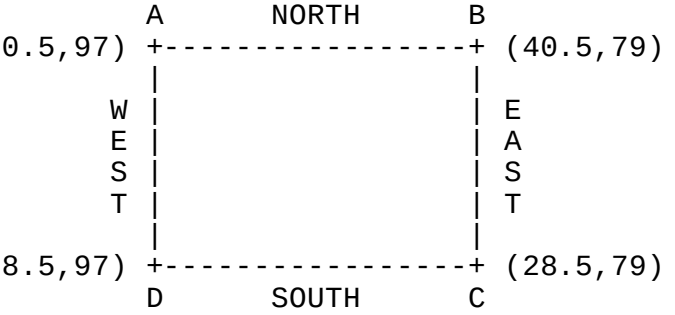
   line 1: the X value of the western most HRAP column
   line 2: the Y value of the southern most HRAP row
   line 3: the maximum number of HRAP columns in the X direction
   line 4: the maximum number of HRAP rows in the Y direction

   These values must be the same as the HRAP grid subset specified in the program PPINIT command DEFINE USER input.

   The following is an example of how to calculate these values:

   Use the northern and southern latitudes and eastern and western longitudes limits from the program PPINIT command DEFINE USER input file to calculate the HRAP coordinates.

An area with a northern latitude limit of 40.5, southern latitude limit of 28.5, eastern longitude limit of 79 and western longitude limit of 97 would look as follows:



Run program find_hrap from the directory $(geo_util)/bin to get HRAP coordinates for the specified latitude and longitude (Usage: find_hrap lat lon):

Command:
find_hrap 40.5 97

Program output:
For latitude value 40.500000 Hrap-row value = 461.371115
For longitude value 97.000000 Hrap-col value = 561.164299

Command:
find_hrap 40.5 79

Program output:
For latitude value 40.500000 Hrap-row value = 566.642018
For longitude value 79.000000 Hrap-col value = 905.490002

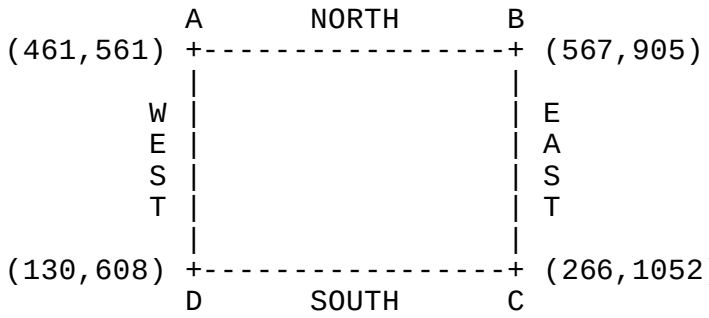
Command:
find_hrap 28.5 79

Program output:
For latitude value 28.500000 Hrap-row value = 266.141192
For longitude value 79.000000 Hrap-col value =
1052.054020

Command:
find_hrap 28.5 97

Program output:
For latitude value 28.500000 Hrap-row value = 130.287075
For longitude value 97.000000 Hrap-col value = 607.695098

The HRAP row and HRAP column values for the HRAP grid subset would be:



The values in the file coord_<user_id>.dat would be:

line 1: 561
line 2: 130
line 3: 491 (1052-561)
line 4: 437 (567-130)

4. **Create the file used for displaying forecast area boundary**.

Create the file rfc_boundary.dat in the ascii directory. 1/

For the following example of a simple square forecast area boundary:

the values in the file rfc_bounday.dat would be:

```
xxx xxx -1 5
40.5 97.0
40.5 79.0
28.5 79.0
28.5 97.0
40.5 97.0
```

Run create_bas_bound from the directory $(geo_util)/bin to create the file rfc_boundary.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound rfc_boundary.dat rfc_boundary.bin
```

5. **Create the file used for displaying basin boundaries**.

Run program PPINIT with the following input:

```
@DUMP PUNCH BASIN ALL
@STOP
```

Move the punch output generated in file ppinit_pun.yyyymmdd.hhmmss to the ascii directory with the filename map_basin.orig.

Run create_bas_bound from the directory $(geo_util)/bin to create the file map_basin.dat in the ascii directory:

```
create_bas_bound
```

Run create_bas_bound from the directory $(geo_util)/bin to create map_basin.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound map_basin.dat map_basin.bin
```

6. **Create the file used for displaying county boundaries**.

Create the file county.bin in the ascii directory. 1/

Run create_bas_bound from the directory $(geo_util)/bin to create the file county.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound county.dat county.bin
```

7. **Create the file used for displaying county warning area boundaries**.

Create the file cwaus.bin in the ascii directory. 1/

Run create_bas_bound from the directory $(geo_util)/bin to create the file cwaus.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound cwaus.dat cwaus.bin
```

8. **Create the file used for displaying flight lines**.

Create the file flights.bin in the ascii directory. 1/

Run create_bas_bound from the directory $(geo_util)/bin to create the file flights.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound flights.dat flights.bin
```

9. **Create the file used for displaying Forecast Groups boundaries**.

Create the file fg_basin.bin in the ascii directory. 1/

Run create_bas_bound from the directory $(geo_util)/bin to create the file fg_basin.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound fg_basin.dat fg_basin.bin
```

10. **Create the file used for displaying forecast points**.

Create the file forecastpt.dat in the ascii directory. The format of this file is as follows:

```
                                    Column
1       10        20        30        40        50        60        70
----+----+----+----+----+----+----+----+----+----+----+----+----+----+
name_of_forecast_point_1 river_name       id1   lat1     lon1
name_of_forecast_point_2 river_name       id2   lat2     lon2
PEC R-BLANCHARDVILLE     NC               BLBNC 41.57    107.91
   .
   .
   .
name_of_forecast_point_N river_name       idN   latN     lonN
```

where name (A24) is in columns 1-24
river (A16) is in columns 26-41
id (A6) is in columns 43-48 (this is what is displayed)
lat (F8.4) is in columns 49-56 (decimal degrees)
lon (F8.4) is in columns 58-65 (decimal degrees)

11. **Create the file used for displaying rivers**.

Create the file river.dat in the ascii directory. 1/

The following is an example of a order 3 river with 5 lat/lon pairs and an order 2 river with 7 pairs of points:

```
xxx xxx 3 5
40.5 97.0
39.8 93.6
38.4 88.1
37.1 85.0
36.7 82.4
xxx xxx 2 7
35.5 87.0
35.3 82.6
36.1 79.1
35.9 77.0
36.0 76.3
36.2 75.8
36.3 74.4
```

Run create_bas_bound from the directory $(geo_util)/bin to create the file river.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound river.dat river.bin
```

12. **Create the file used for displaying state boundaries**.

Create the file state.bin in the ascii directory. 1/

Run create_bas_bound from the directory $(geo_util)/bin to create the file state.bin in the directory $(geo_data)/<user_id>/binary:

```
create_bas_bound state.dat state.bin
```

13. **Create two files used for displaying towns**.

Create the files town.dat and town_zoom.dat in the ascii directory.

The cities in the town.dat file are displayed all the time while the cities in the town_zoom.dat file are only displayed when you zoom in. The format for these files is as follows:

```
name1 lat1 lon1
name2 lat2 lon2
   .
   .
nameN latN lonN
```

Notes:

1/ The values in the file are:

```
line  1             : id name order npts
lines 2 thru npts+1 : latitude longitude

where id     is the identifier (maximum 8 characters)
      name   is the name (maximum 20 characters)
      order  is the order value used to determines when the
             item is displayed on the screen (-1 if no order):
                3 = the item shows up in initial display and
                    all levels of zoom
                2 = the item is displayed zooming in
      npts   is the number of latitude/longitude points
```

The values must be in a clockwise order and have closure (the first value and last value must be the same).

The more values the better the resolution.