The Mapserver, Terascan, and Ice Observations Form project software was written for and tested using the previous SCS Data System real-time data format on the USCGC Healy. This software was intended for use on the USCGC Polar Star, but was never updated to use the newer real-time data format from the USCGC Polar Star data systems. This project was halted per the request of USCG's Dave Cohoe (Dave.Cohoe@uscg.mil).

J. Scannell (anstett@ucar.edu)
NCAR/EOL
June 30th, 2013

Additional Mapserver Information

All of the information needed for various tasks for Mapserver is included in the GISMappingProject.pdf training documentation. There is a general section on Mapserver and there is a section on common tasks in Mapserver. There is information on the Mapserver directories and files, and web sites for obtaining more information located at the end of the training documentation. Additionally, there is the msice-exercise.pdf training documentation that includes exercises on adding ice imagery to Mapserver.

Add layers to Mapserver

1) Edit layers.inc.js file and add the code to display the layer in the column on the left side of the Mapserver web page.
2) Add information about the layer to the map file (polarstar.map).
3) Test the layer in Mapserver.
4) Add template html files for the layer.

NOTE: Only make a few changes at a time to the map file, since it makes it easier to debug problems.

Ship Tracks

NOTE: The creation of the ship tracks is based on the previous SCS Data System real-time data format. The programs will have to be updated to use the newer real-time data format on the USCGC Polar Star data systems.

There are programs that use the real-time ship data to create ship tracks and one-minute shapefiles for displaying in mapserver. Once these programs are set up properly, the ship track and position can be updated automatically in mapserver by running the programs as a cron job.

The tracks are created in the temp directory (/mnt/mapserver/data/polarstar/data/ship/temp) and once the program has completed, the resulting tracks are copied to the appropriate mapserver ship directory (/mnt/mapserver/data/polarstar/data/ship). A copy of the last complete track from the previous day is kept in the complete directory (/mnt/mapserver/data/polarstar/data/ship/complete), so the program only has to add the information for the current day to the ship track.

partialtrack.pl – Runs every 10 minutes to update the ship track and position. First, the program checks to see if there are track shapefiles in the complete directory. If not, new track shapefiles are created. If so, the current track shapefiles are copied to the temp directory. Then, createtrack.pl is run for the current day. Lastly, the shapefiles are converted to the Polar Stereographic projection and the track shapefiles are moved to the ship directory. When the ship track is viewed in mapserver, it will be updated for the last time the partialtrack.pl program was run.

completetrack.pl – Runs once per day to update the ship track and position for the entire previous day. First, the program checks to see if there are track shapefiles in the complete directory. If not, new track shapefiles are created. If so, the current track shapefiles are copied to the temp directory. Then, createtrack.pl is run for the entire previous day and the information is added to the current track shapefiles. Lastly, the shapefiles are converted to the Polar

Stereographic projection and the track shapefiles are moved to the ship directory.
If no date is specified on the command line when completetrack,pl is run, the entire previous day will be created in the track shapefiles. If a date is specified, the track shapefiles for that specific date will be run. This allows a track to be recreated from the start of the cruise, if necessary.

createtrack.pl – This program is run from the partialtrack.pl and completetrack.pl programs. The program takes the real-time data from the ship data files and adds the information for either the current day or the entire previous day to the track shapefiles.

NOTE: The track shapefiles are converted to Polar Stereographic projection, since Mapserver has a known problem displaying lat, lon projected files in the polar regions. If a lat, lon projected shapefile is being displayed on the map in the polar regions, when the map is zoomed in or out, or panned, the layer may disappear from the map.

Track shapefiles that are created:
track.dbf
track.shp
track.shx
track_stere.dbf
track_stere.prj
track_stere.shp
track_stere.shx
track.* are the lat, lon projections of the current track information.
track_stere.* are the Polar Stereographic projections of the current track information.

track_current.dbf
track_current.shp
track_current.shx
track_current_stere.dbf
track_current_stere.prj
track_current_stere.shp
track_current_stere.shx
track_current.* are the lat, lon projections of the current ship position.

track_current_stere.* are the Polar Stereographic projections of the current ship position.

NOTE: All the track shapefiles include one-minute data for selected variables.

Information on adding the Zoom to Ship button

The html/polarstar/mapper.html file needs to have the following three lines uncommented:

1) Near this line:
```
var states = new Array(
```
uncomment this line:
```
//  'ship',
```

2) Near this line:
```
var alttext = new Array(
```
uncomment this line:
```
//  'Zoom to Ship',
```

3) Near this line:
```
var actions = new Array();
```
uncomment this line:
```
//actions[i++] = 'myMap.zoomShip(); swap(\'ship\', \'ship_off\');';
```

The html/polarstar/javascript/mapserver.js file needs the Map_zoomShip function edited:

1) Change the layer name to the current ship layer:

```
    // Need to make sure current track layer is on since this is the
    // layer we will query on.
    if(notinlayer(zoomlayers,"polarstar_track")) {
        zoomlayers.push("polarstar_track");
    }
```

2) Change the following query to match the shapefile for the current ship layer. &qitem needs to be changed to the name of a field in the

shapefile and &qstring needs to be changed to a value that will be found in the shapefile for the qitem field.

```
   // qitem is something to query in qlayer,
   //  qstring is a value of qitem that is guaranteed to return a result;
   //  this is a kludge to get refresh a valid URI to reload
       var options = '&qitem=ID'
               +'&qstring=1'
               +'&qlayer=polarstar_track'
               +'&mapxy=shape'
               +'&scale='+scale
               +'&program='+this.mapserver
               +'&map_size='+this.mapw+'+'+this.maph
//             +'&map_style=normal'
               +'&layers='+zoomlayers.join('+')
               +'&map_web=HEADER+loader.html';
```

When the ship track shapefiles are created (see Ship Tracks above), there should be a field included named ID with values that start at 1 and increase for each point in the shapefile. If this is done, then there will be a qitem and qstring to query that will be guaranteed to return a result.