MeteoInfo User Guide

Version 1.0

Introduction

MeteoInfo is freely available software designed to view and analyze meteorological and spatial data interactively. Some GIS functions were developed from ground level. It was developed with C# in the Microsoft .Net environment. MeteoInfo may be run in Windows with .NetFramework 3.5, or in UNIX like systems with Mono, an open source implementation of the .NET Framework. MeteoInfo can also be run automatically using MeteoInfo scripting with the IronPython language. The main functions are packed in the MeteoInfo class library, which could be used to conveniently develop the software. This document is based on the version 1.0. Any suggestion is welcome.

The MeteoInfo software can be used and distributed freely. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and no infringement.

Author: Dr. Yaqiang Wang
Email: yaqiang.wang@gmail.com, wangyq@cams.cma.gov.cn
Chinese Academy of Meteorological Sciences, CMA

Download and Install MeteInfo

MeteInfo could be downloaded freely from the website http://www.esnips.com/web/MeteoInfo. The .rar file could be unzipped to an executable file. Run it to start the installation process, which is quit easy and you just need to press 'next' button. Under MeteoInfo installation folder, 'Map' directory includes some geographical map data and 'Sample' directory includes some sample meteorological data.

The other file with .zip extension including all files to run MeteoInfo, and it don't need to be installed. Just unzip it in your disk and run the file of MeteoInfo.exe to start MeteoInfo. This file could be used both in Windows and Linux system. For Windows, you need to copy 'WeatherSymbol.ttf" to system's font folder for weather symbol view.

MeteoInfo must be installed in a Microsoft Windows system with .Net Framework 3.5, which could be downloaded freely from Microsoft website if your computer has no it.

MeteoInfo could also be run in other operational system such as Linux with support of Mono. NetCDF file can not be opened under Mono at present.

1. Install mono if your system has no it. Download Mono on the website http://www.go-mono.com/mono-downloads/download.html. Use the binary installation file if it is available, or you need to download the source codes and install it as following steps.

```
# ./config
# make
# make install
```

2. Install ‘WeatherSymbol.ttf’ font.

cd /usr/share/fonts

Copy the font file to above folder

mkfontscale

mkfontdir

fc-cache

3. Run MeteoInfo.

mono MeteoInfo.exe

Supported Data

Geographic Map Data

The following geographic map data were supported by MeteoInfo.

- ESRI shape file with point, polyline or polygon shape type.
- Geographic map data of GrADS.
- Geographic map data of MICAPS.
- wmf map data. (Can be created by ‘Output Map Data’ function in MeteoInfo)

A shapefile is a digital vector storage format for storing geometric location and associated attribute information. This format lacks the capacity to store topological information. Shapefiles are simple because they store primitive geometrical data types of points, lines, and polygons. These primitives are of limited use without any attributes to specify what they represent. Therefore, a table of records will store properties/attributes for each primitive shape in the shapefile. Shapes (points/lines/polygons) together with data attributes can create infinitely many representations about geographical data. Representation provides the ability for powerful and accurate computations.

While the term "shapefile" is quite common, a "shapefile" is actually a set of several files. Three individual files are mandatory to store the core data that comprises a shapefile: ".shp", ".shx", ".dbf", and other extensions on a common prefix name (e.g., "lakes.*"). The actual *shapefile* relates specifically to files with the ".shp" extension, but alone is incomplete for distribution, as the other supporting files are required.

Some resources for freely available shapefiles are:

- National Weather Service Shapefile Database
- National Atlas
- DIVA-GIS

Meteorological Data

Following meteorological data were supported at present. More data format will be supported in future.

- NetCDF grid data
- GRIB edition 1 and 2 data
- GrADS binary grid and station data
- ARL packed meteorological data
- HYSPLIT model output data: concentration, particle and trajectory
- METAR data (special for the cycle METAR data of NOAA, http://weather.noaa.gov/weather/metar.shtml)
- NOAA ISH data
- Lon/Lat Station ASCII data
- MICAPS 1, 3, 4, 7, 11 and 13 format data
- ESRI ASCII Grid data
- Surfer ASCII Grid data
- AWX data

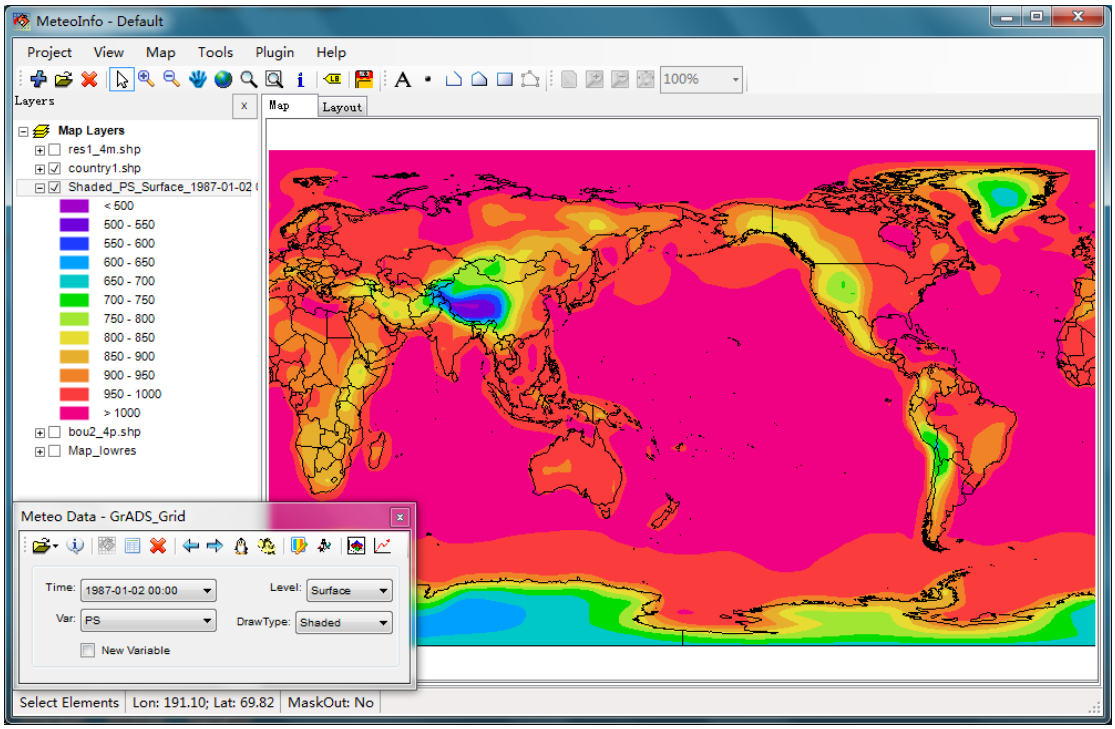
Image Data

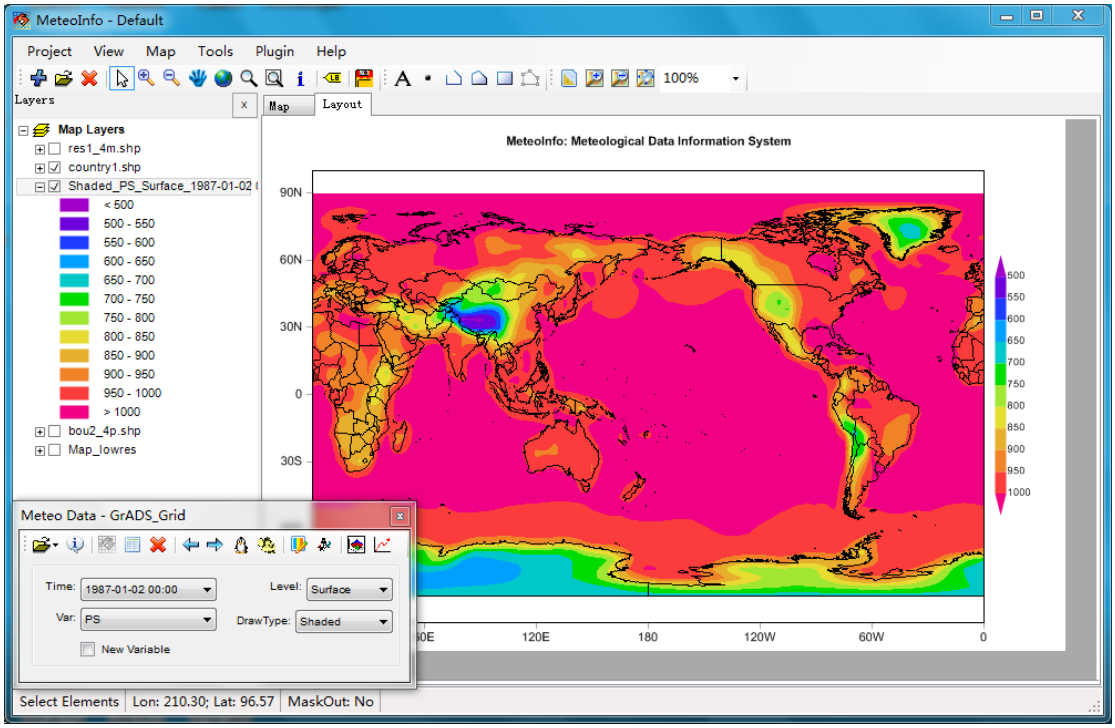
The image data supported by MeteoInfo:

- Bmp
- Gif
- Jpg
- Tif
- Png

Main form

The main form of MeteoInfo contains menus, tools, status, a layer control, a map view and a layout view. The contents of the layers will be showed in map view. In layout view also legend, title, illustration and grid labels could be showed.





Layers and project file

In MeteoInfo, the data were arranged with layers. All layers saved as files and the setting of MapView and Layout could be saved as a project file (.mip). The layer created from meteorological data only exists in memory. It can be saved as a shape file and then could be saved

as a layer in the project file.

When MeteoInfo started, a default project file ‘Default.mip’ will be loaded.

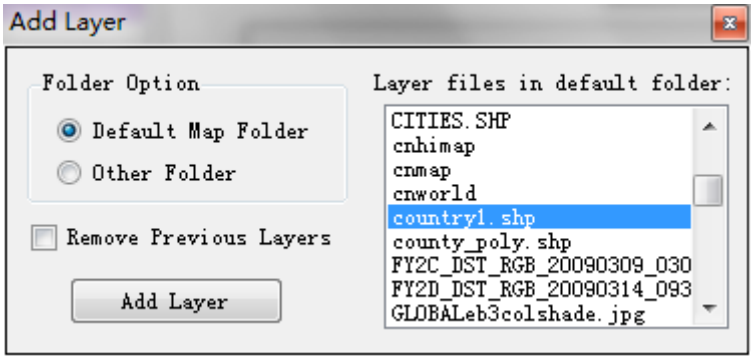
Using layers

There are three kinds of layer in MeteoInfo.

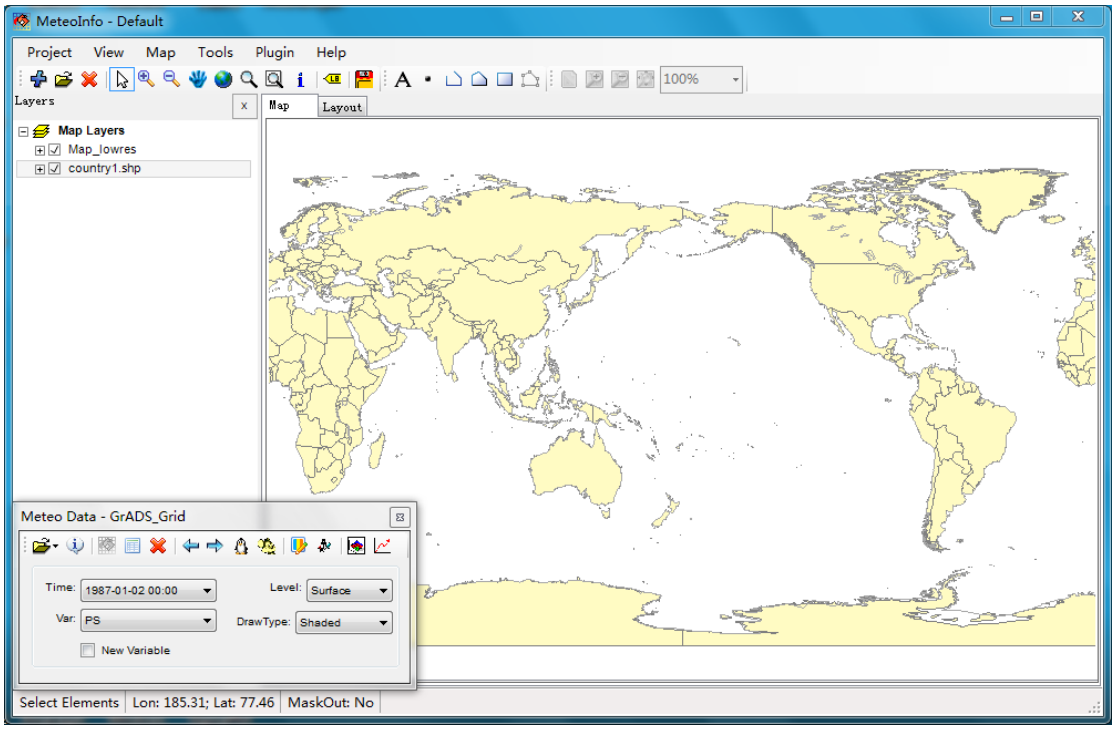
- Vector layer (Point, Polyline, Polygon), has attribute data
- Image layer (geo-location setting, world file)
- Raster layer (Grid data)

Add layer

Press ‘Add Layer’ button  to open the ‘Add Layer’ dialog. All the layer data files in the default map folder (‘Map’ folder under software installation path) were showed in right list box. You could select one and open it by pressing ‘Add Layer’ button. Choose ‘Other Folder’ and then press ‘Add Layer’ button to open a map data file which is not in the default folder. Please pay attention the option ‘Remove Previous Layers’. All previous loaded layers will be removed when you add a new layer with the option checked.



In this case, we select to open ‘country1.shp’ file with ‘Remove Previous Layers’ option unchecked. ‘country1.shp’ is a polygon shape file, so it is opened as a colored map. The default color is light yellow.



Edit Layer Properties

Double click the layer ‘country1.shp’ to open its ‘Layer Set’ dialog. Some information of the layer such as ‘FileName’, ‘LayerType’ were shown as ‘Layer Property’ which is read only. You could change the parameters in ‘Layer Set Edit’ section.

Vector layer properties:

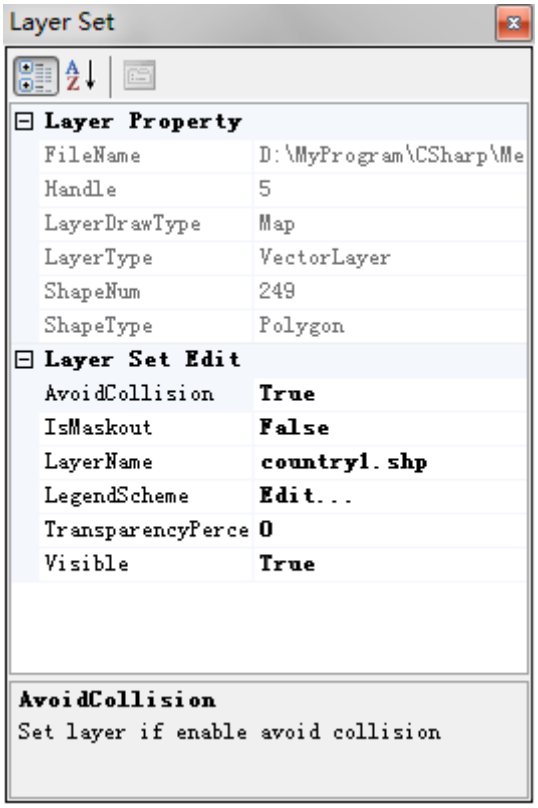
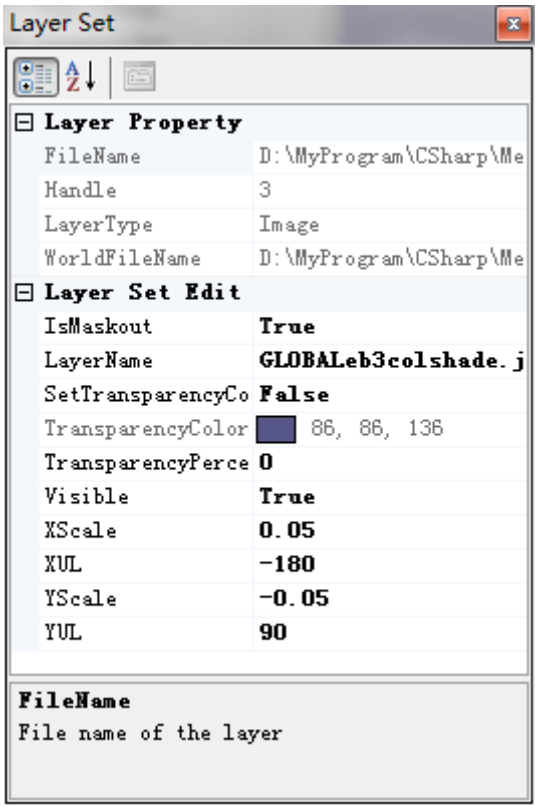
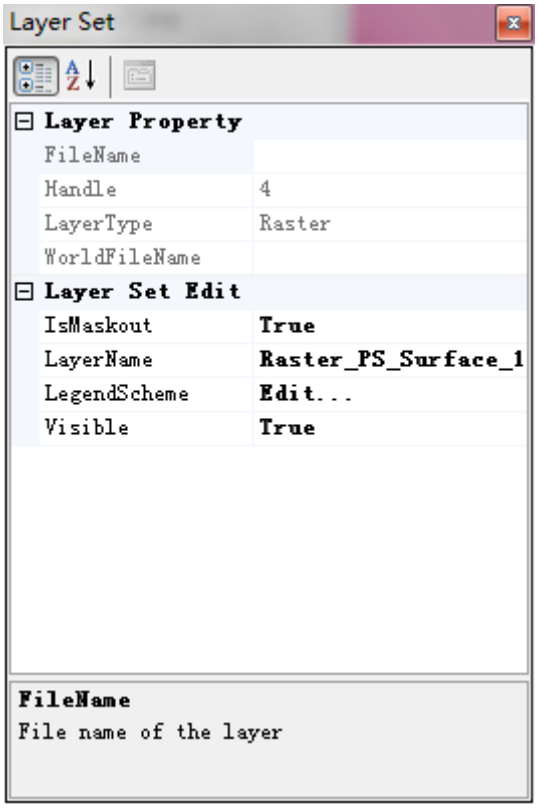


Image layer properties:

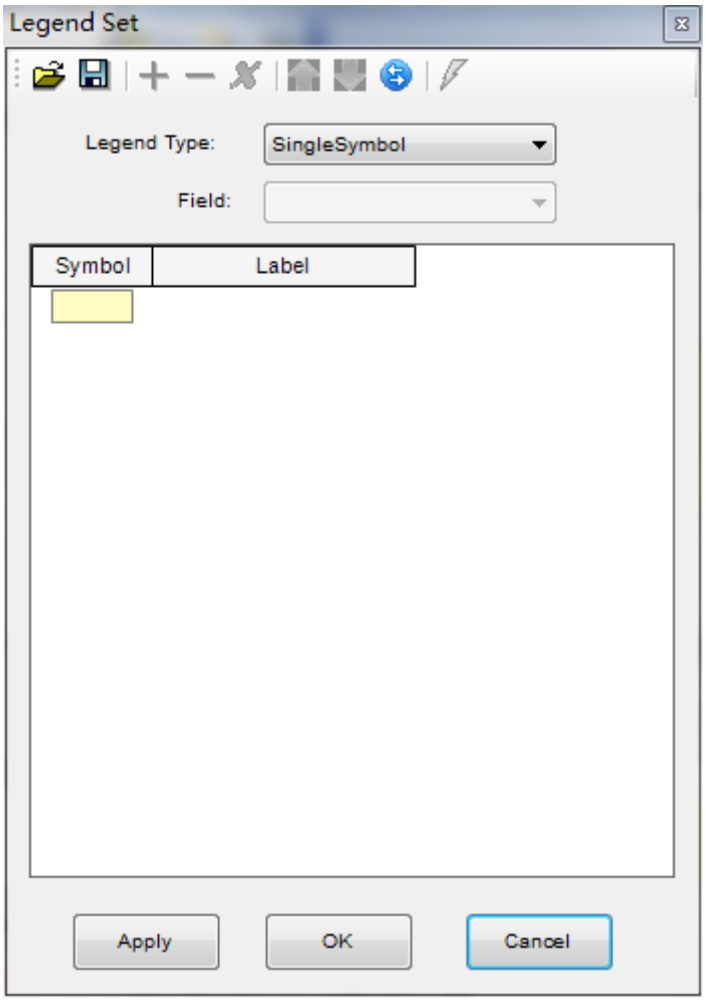


Raster layer properties:

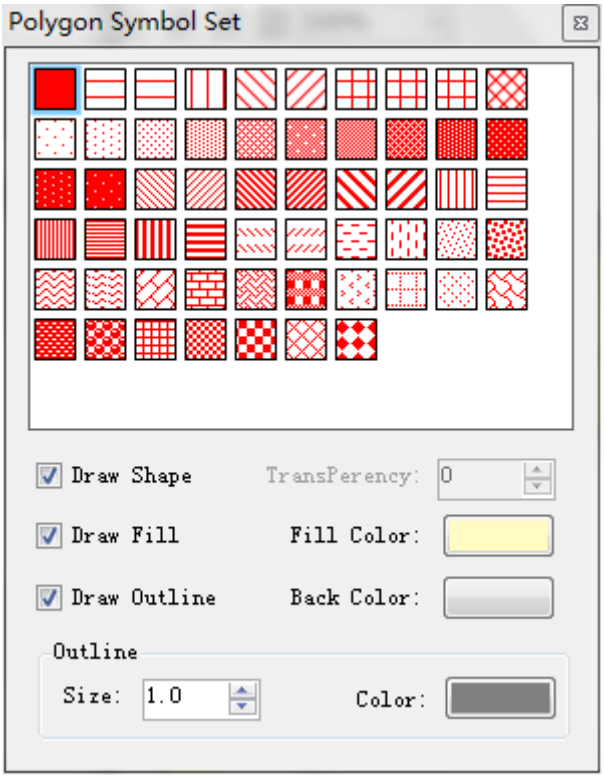


Legend Scheme

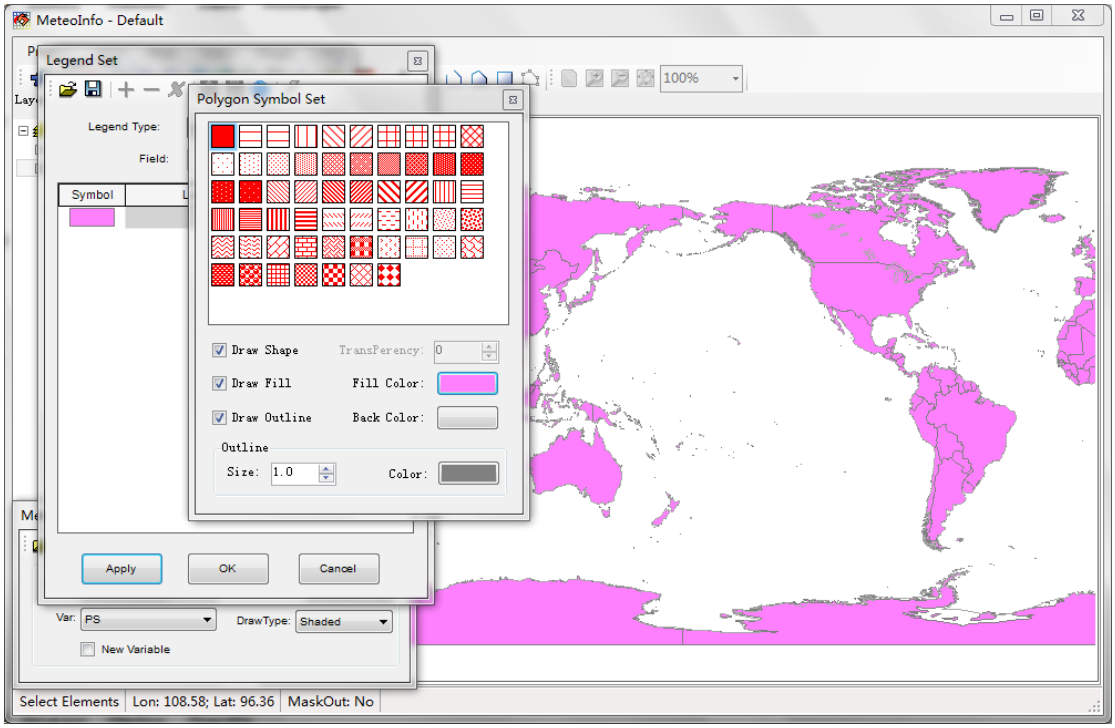
The visualization behavior of a layer is base on its legend scheme. The legend scheme of the layer could be edited with ‘Legend Set’ dialog which could be opened by pressing ‘Edit…’ of ‘LegendScheme’ and the run time button on right side.



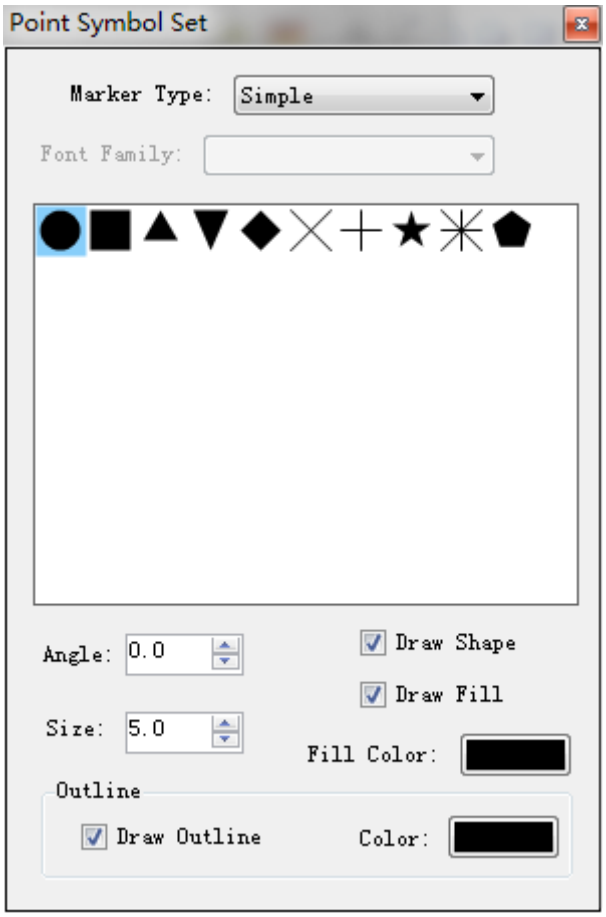
Double click the colored rectangle under ‘Symbol’ column to open a ‘Symbol Set’ dialog.

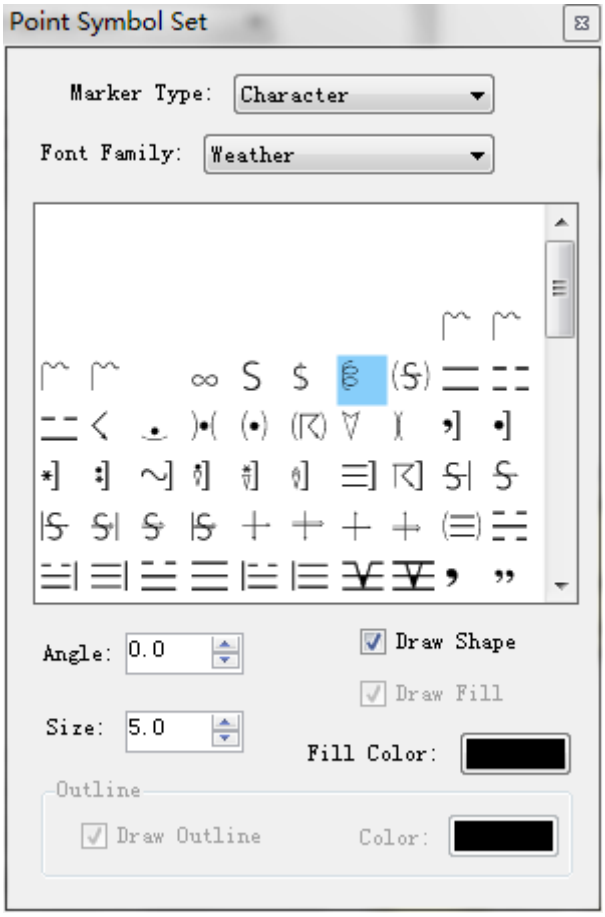


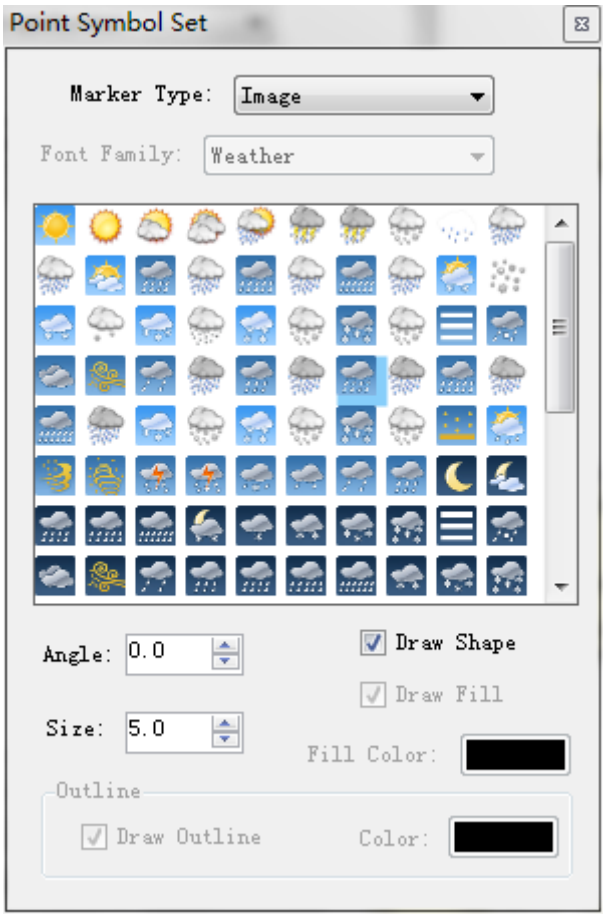
Now change ‘Fill Color’ and press ‘Apply’ button in ‘Legend Set’ dialog, the color change is viewed in map window.



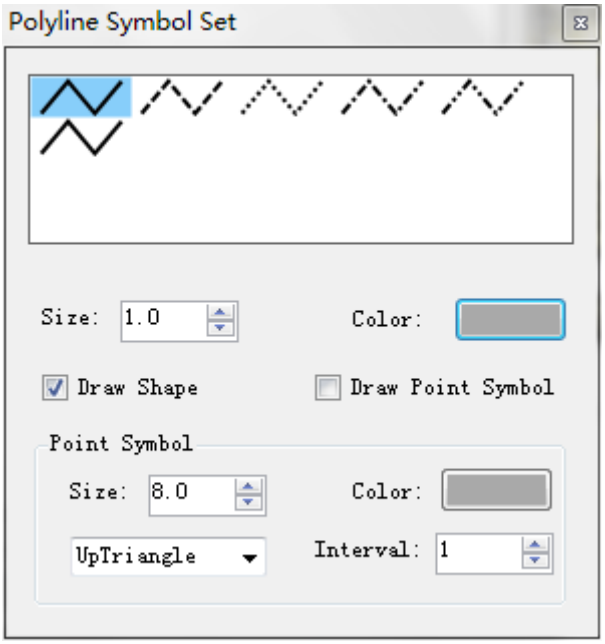
Point symbol setting dialog is showed bellow. The marker types are ‘Simple’, ‘Character’ and ‘Image’.


Point Symbol Set
Marker Type:
Simple
Font Family:
Angle:
0.0
Size:
5.0
Draw Shape
Draw Fill
Fill Color:
Outline
Draw Outline
Color:


Point Symbol Set
Marker Type:
Character
Font Family:
Weather
Angle:
0.0
Size:
5.0
Draw Shape
Draw Fill
Fill Color:
Outline
Draw Outline
Color:



Polyline symbol setting dialog is showed bellow.

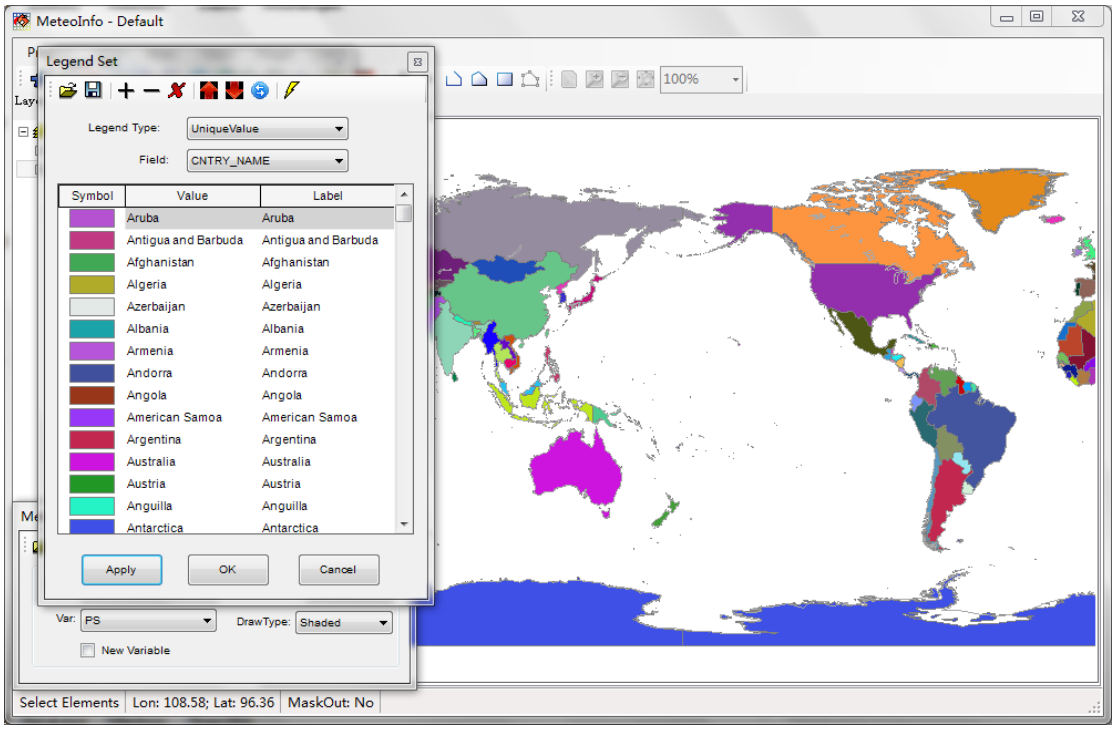


There are three legend types in MeteoInfo.

- ‘SingleSymbol’. All shapes of the layer are set in one same symbol.
- ‘UniqueValue’. Each shape of the layer has its own symbol.

✧ ‘GraduatedColor’. The shapes are grouped according their values an each group has its own symbol.

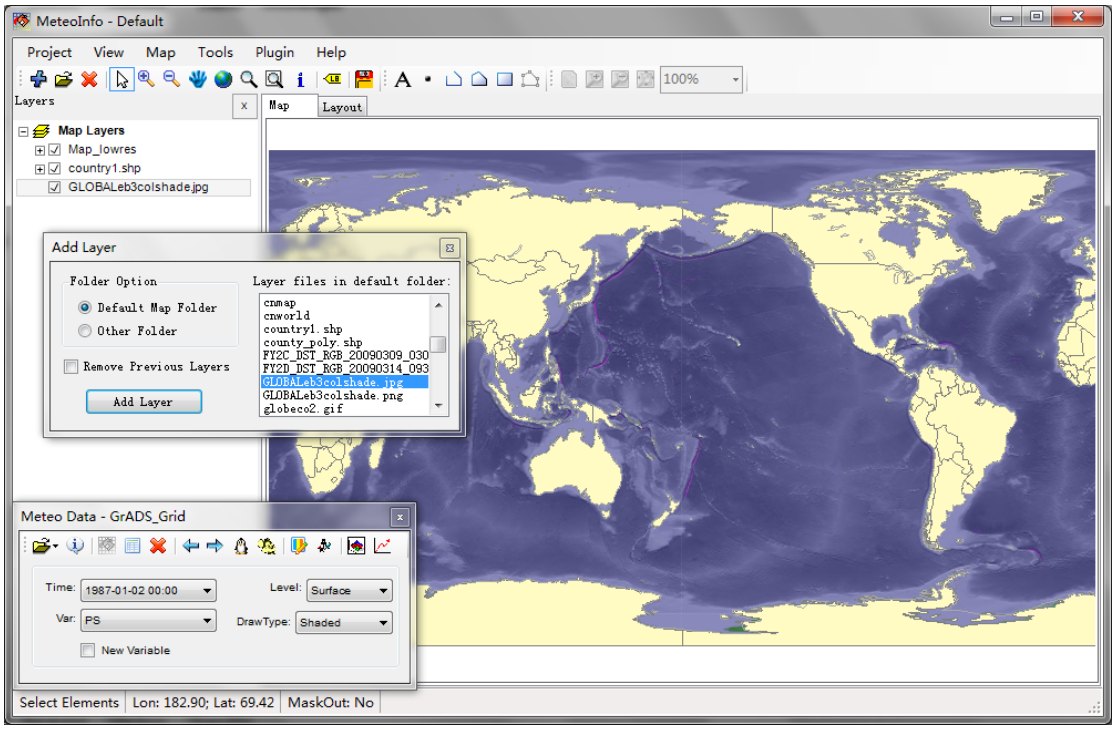
For example, after you change the legend type to ‘Unique Value’ and select the ‘Field’ to ‘CNTRY_NAME’. After press ‘Apply’ button, the map data will show as follow figure.



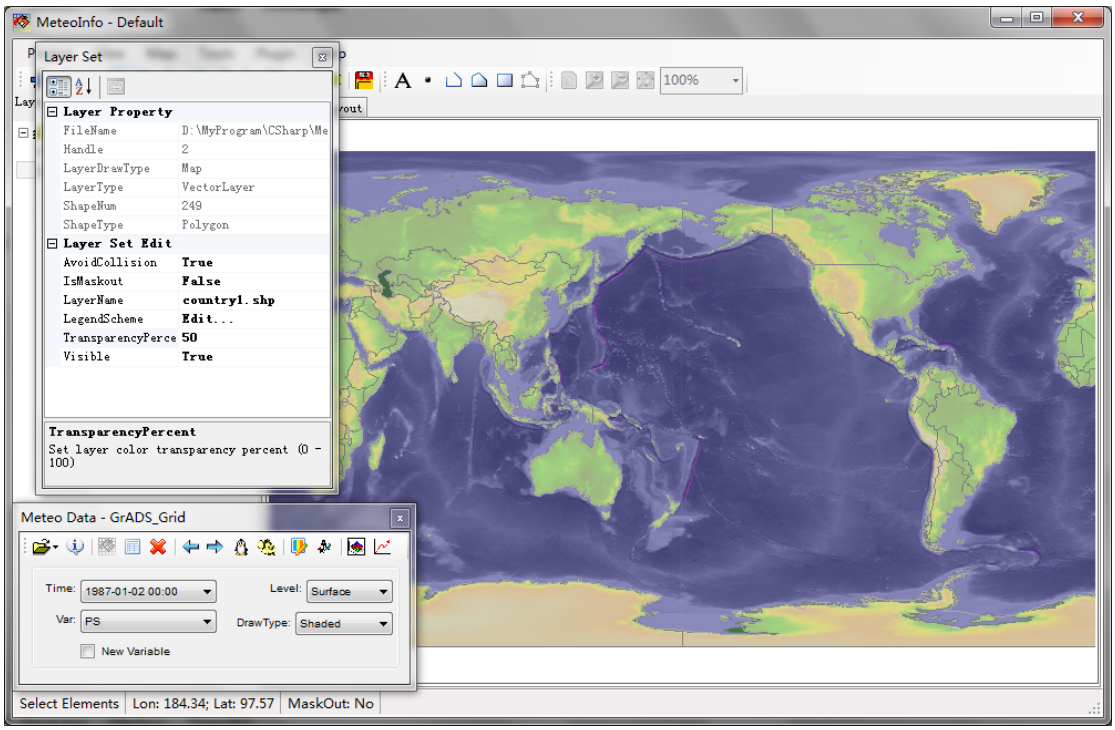
The legend setting could be saved as a legend file with .lgs extension. And the legend file could be loaded also. Under ‘UniqueValue’ and ‘GraduatedColor’ types, some tools could be used to add/remove and move the break in the legend set. The legend changes will be saved to the layer only by pressing ‘OK’ button.

Transparency Set

Add an image layer ‘GLOBALeb3colshade.jpg’ and the image on the land was hided by ‘country1.shp’ layer.



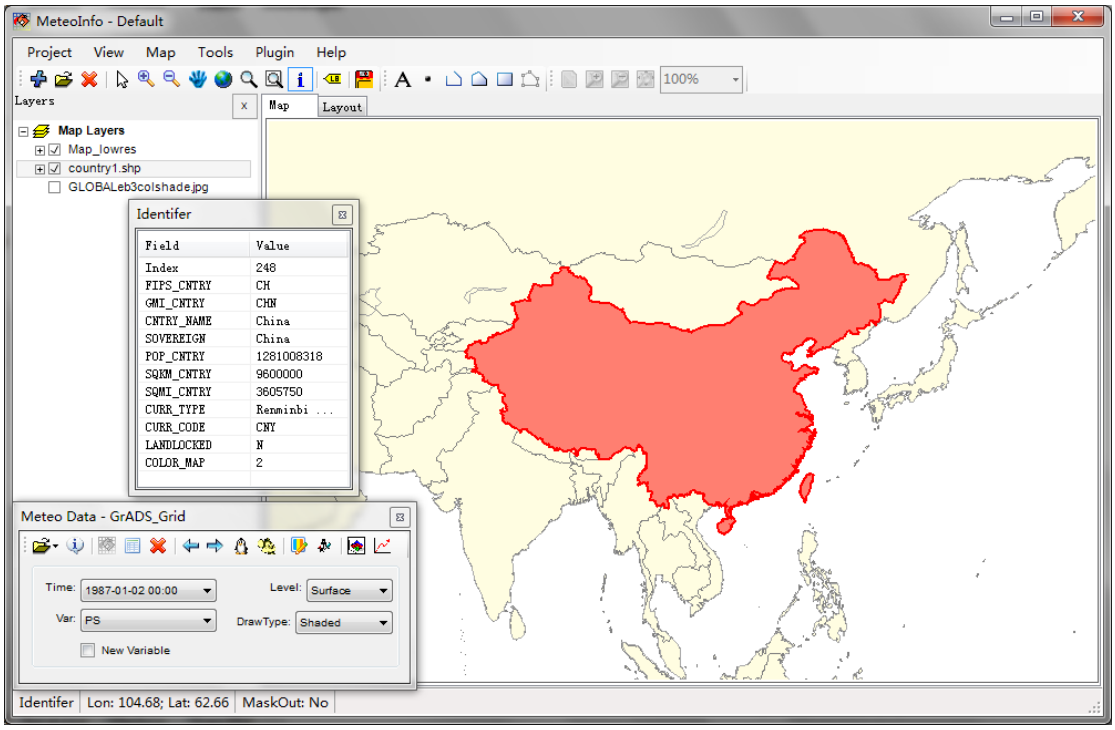
Set the ‘TransparencyPercent’ of the layer ‘country1.shp’ to 50.



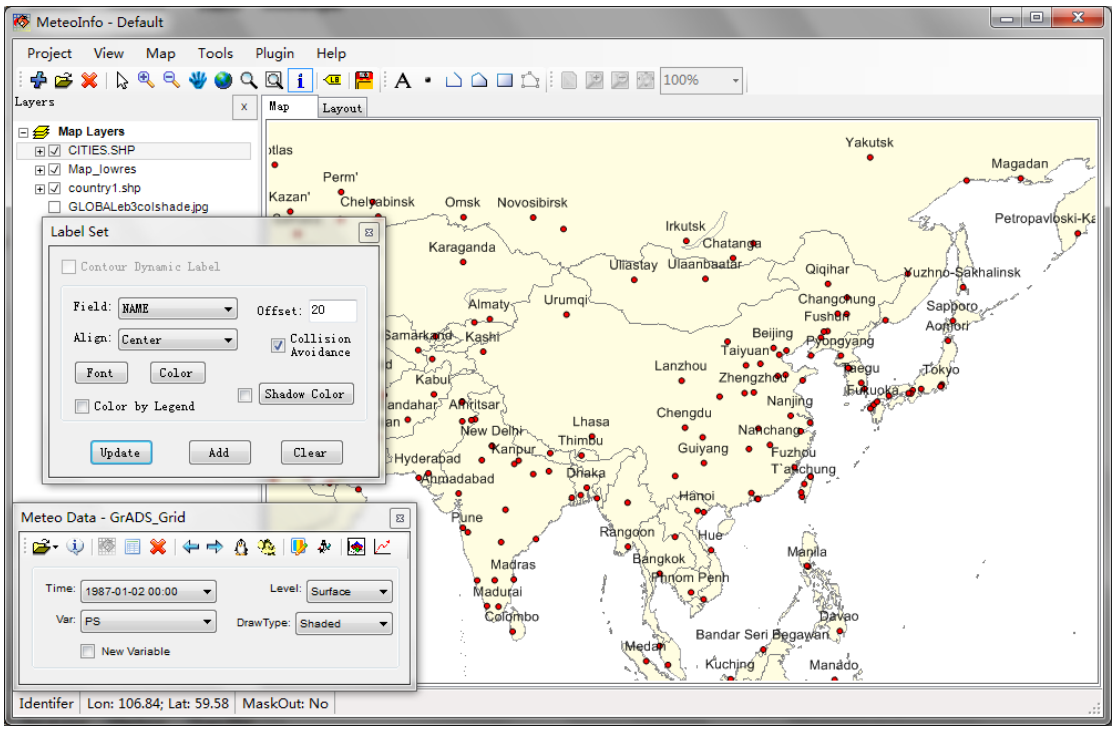
Attribute Data

The vector layer has attribute data. One row data in the attribute table corresponds with one shape of the layer. The attribute data of a shape which is highlighted could be identified using

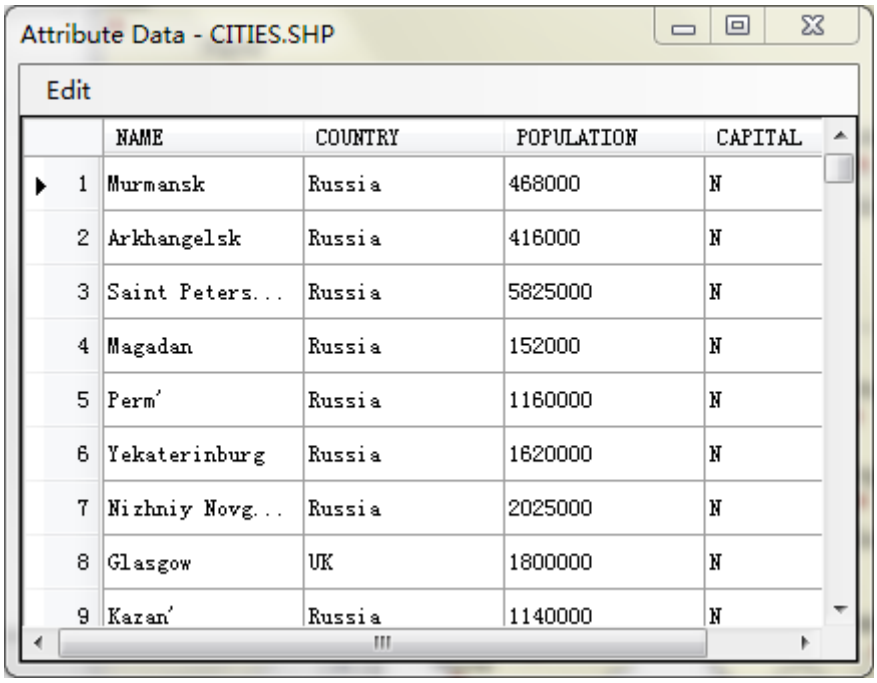
'Identifer' tool button  .



The shapes could be labeled by the attribute data with 'Label' tool  .



The attribute data table could be opened by 'Attribute Data' menu.



	NAME	COUNTRY	POPULATION	CAPITAL
1	Murmansk	Russia	468000	N
2	Arkhangelsk	Russia	416000	N
3	Saint Peters...	Russia	5825000	N
4	Magadan	Russia	152000	N
5	Perm'	Russia	1160000	N
6	Yekaterinburg	Russia	1620000	N
7	Nizhniy Novg...	Russia	2025000	N
8	Glasgow	UK	1800000	N
9	Kazan'	Russia	1140000	N

Customize Image Layer

World File

Images are stored as raster data, where each cell in the image has a row and column number. In order to display images in real-world coordinates, it is necessary to establish an image-to-world transformation that converts the image coordinates to real-world coordinates. This transformation information is typically stored with the image.

This information was stored in a separate ASCII file. This file is generally referred to as the world file, since it contains the real-world transformation information used by the image. World files can be created with any editor. They can also be created using MeteoInfo.

It's easy to identify the world file which should accompany an image file. The first and third characters of the image file's suffix and a final "w" are used for the world file suffix. Therefore, for 'mytown.tif', the world file would be 'mytown.tfw'. For 'redlands.gif', its world file would be 'redlands.gfw'.

The contents of the world file will look something like this:

```
20.17541308822119      -A
 0.00000000000000      -D
 0.00000000000000      -B
-20.17541308822119     -E
424178.11472601280548  -C
4313415.90726399607956 -F
```

When this file is present, MeteoInfo performs the image-to-world transformation.

A = x-scale; dimension of a pixel in map units in x direction
B, D = rotation terms
C, F = translation terms; x,y map coordinates of the center of the upper-left pixel
E = negative of y-scale; dimension of a pixel in map units in y direction

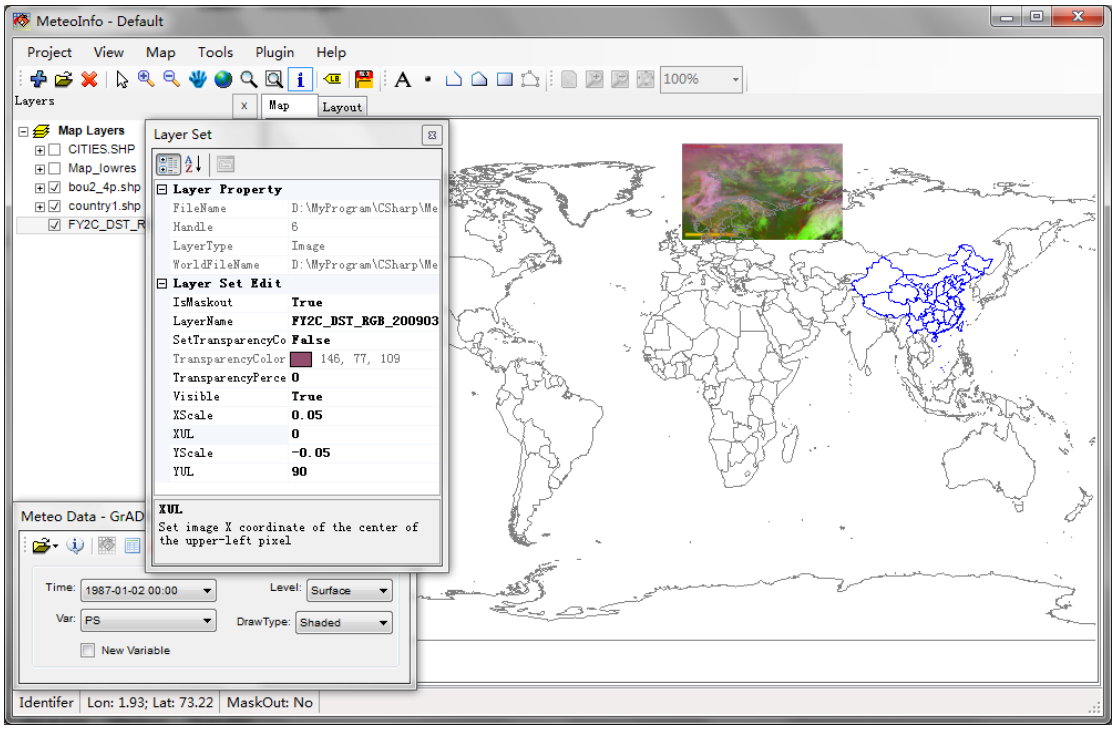
Note: The y-scale (E) is negative because the origins of an image and a geographic coordinate system are different. The origin of an image is located in the upper-left corner, whereas the origin of the map coordinate system is located in the lower-left corner. Row values in the image increase from the origin downward, while y-coordinate values in the map increase from the origin upward.

Note: No rotation terms were considered in MeteoInfo at present.

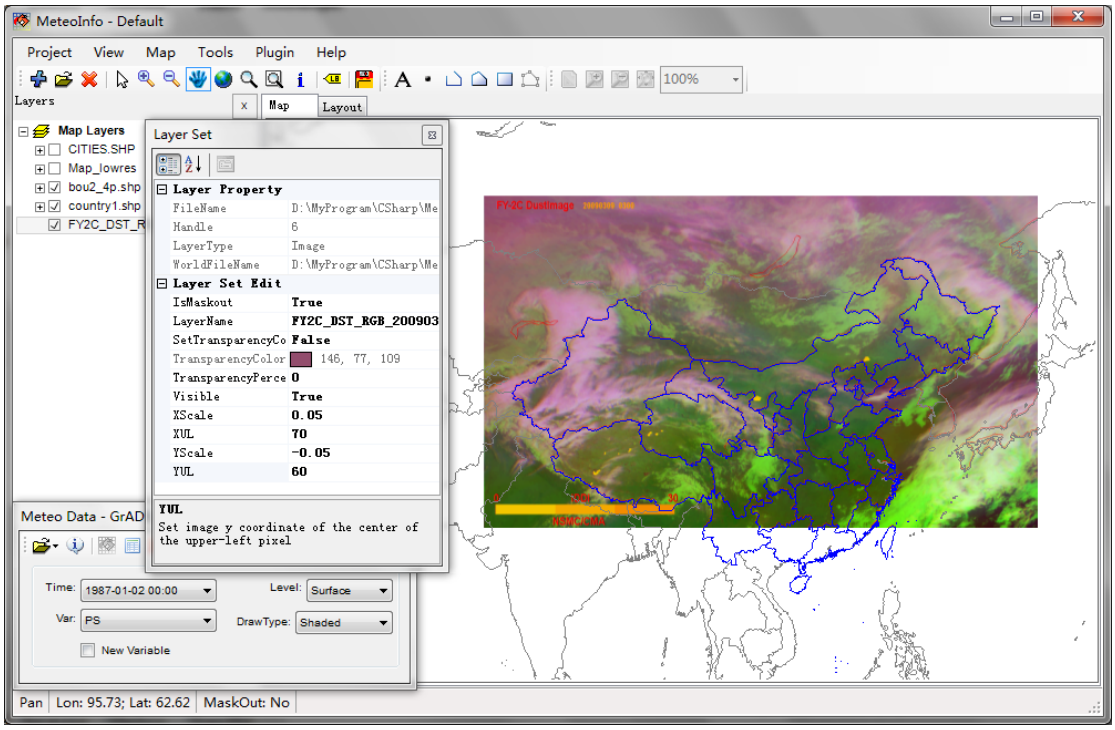
Geo-locate Image Layer

If the image has world file, MeteoInfo will read the transform information and view the image according to the information. If the image data has no world file, MeteoInfo will write a world file automatically. Then you can adjust the parameters to view the image with correct spatial position.

Add an image without world file. Double click the image layer name in 'Layers' area. The transform parameters were showed in 'Layer Set' dialog. Change the value of 'XScale', 'XUL', 'YScale', 'YUL' to adjust the image in the map. Normally you have to try many times, so be patient of it.



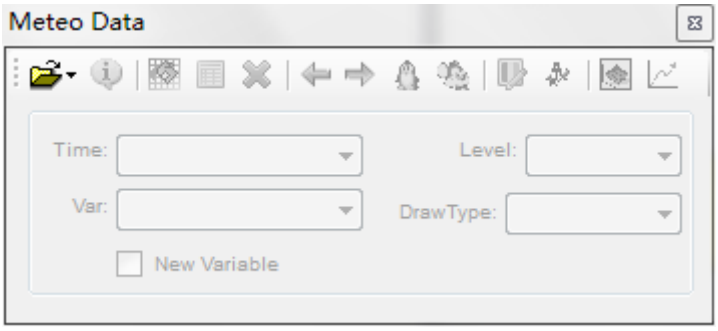
After adjust correctly.



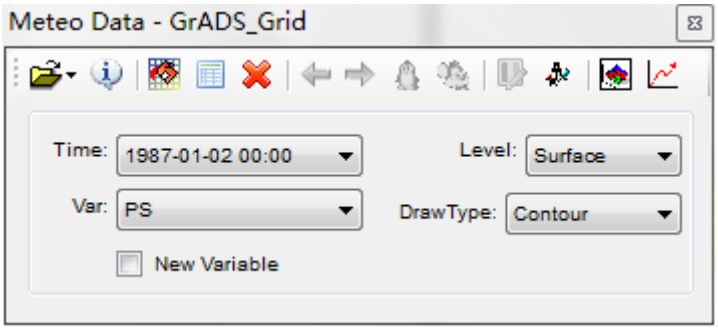
Using Meteorological Data

Open Meteorological Data

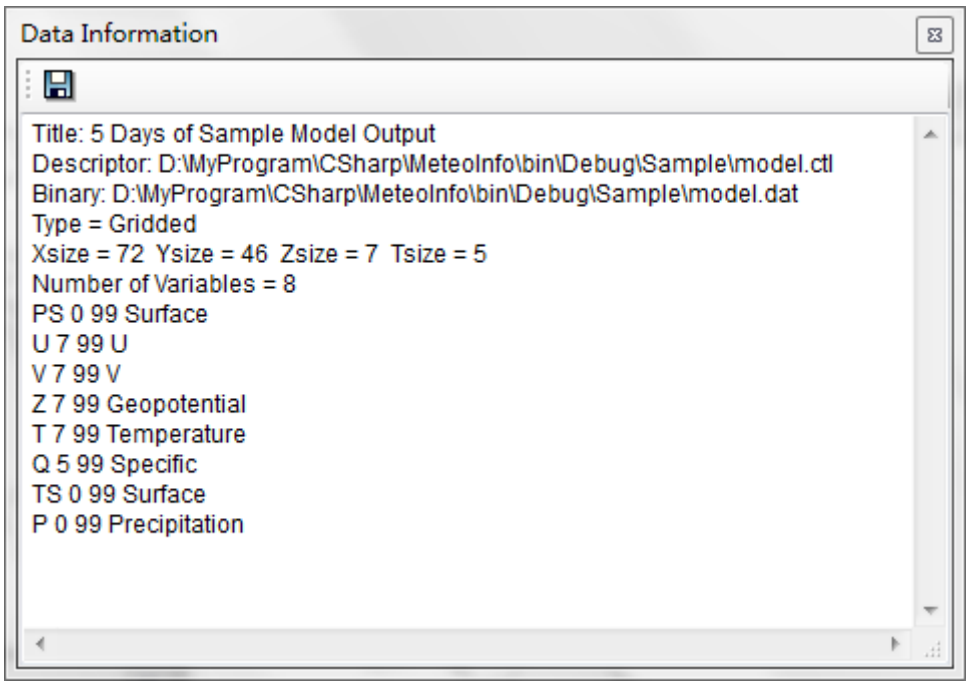
'Meteorological Data' dialog is opened when MeteoInfo is started. Press 'Open Data' button will reopen the dialog if it is closed.



The supported meteorological data formats were listed under 'Open Data' button. Click one of them and select one meteorological data file. Some data information will be showed such as times, variables and levels.



The more detailed information of the data could be viewed by pressing 'Show Data Information' button.



Plot Gridded Data

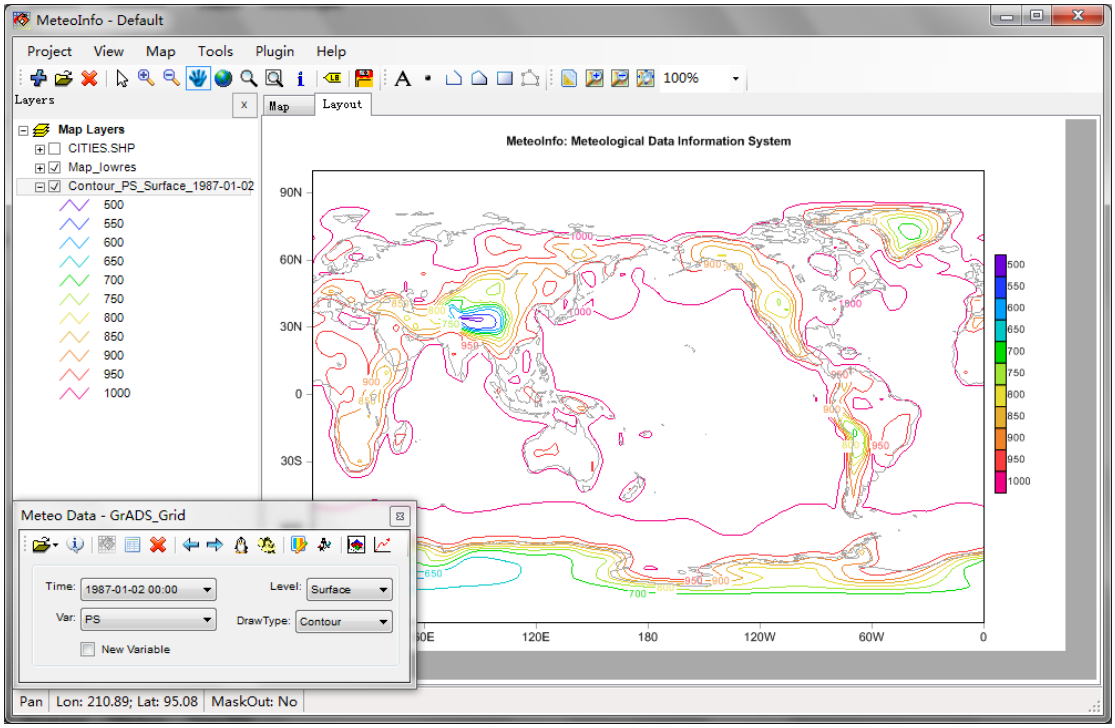
Gridded data could be plot as following graphic types.

- ✧ Contour
- ✧ Shaded
- ✧ Grid Fill
- ✧ Grid Point
- ✧ Vector
- ✧ Barb
- ✧ Streamline
- ✧ Raster

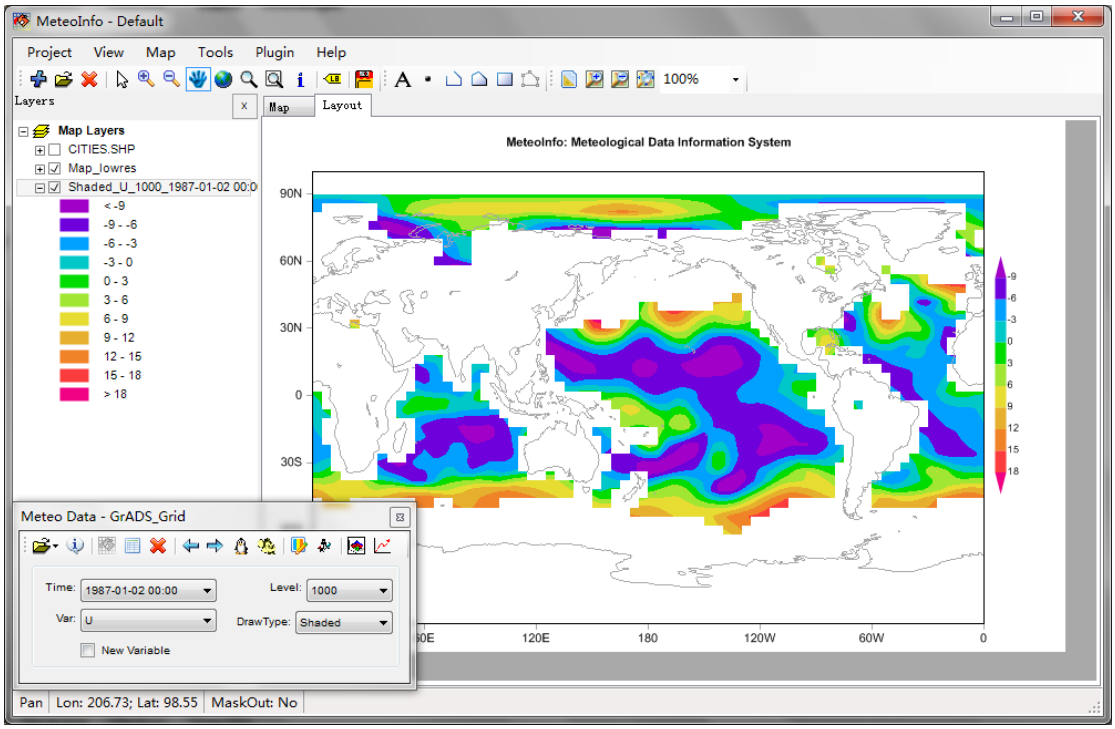
The following pictures were plotted with sample data 'model.ctl' and 'model.dat'.

Contour and Shaded Graphics

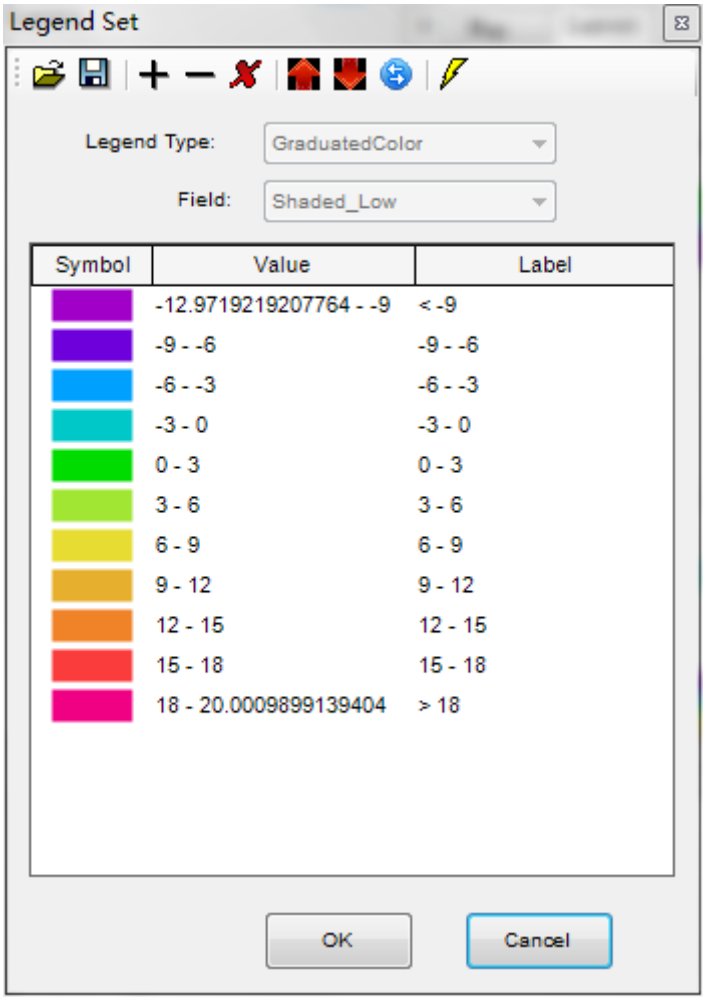
Set ‘Time’, ‘Level’, ‘Var’ items and select ‘Draw Type’ to ‘Contour’, and then click ‘Draw Data’ button. The software will trace the contour lines from the gridded data according to the setting. A layer was created including all the drawing shapes, and was added in ‘Layers’ window. The layer name was set automatically with draw type, various, level and time.



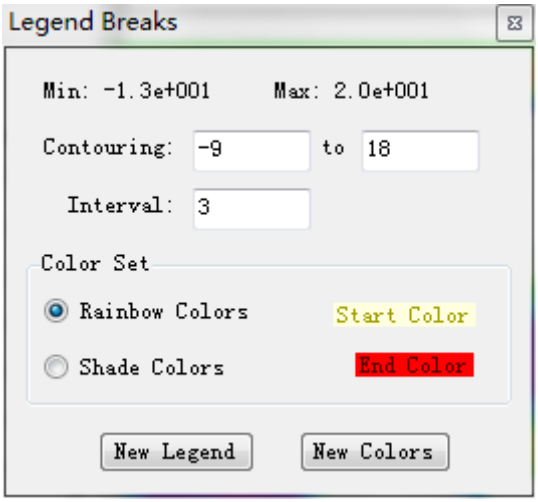
Shaded graphic is similar with contour graphic but with color filled.



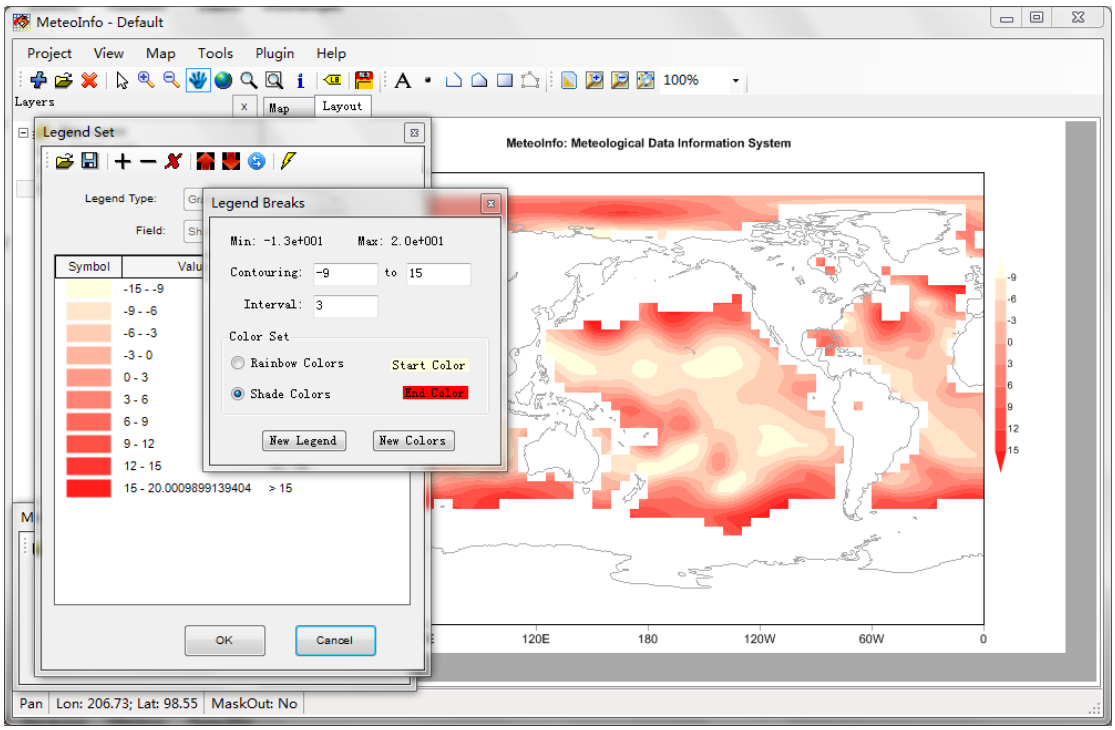
The contour values will be calculated automatically by the software and the colors will be set as ‘Rainbow’ by default. The legend setting could be edited in ‘Legend Set’ dialog which could be opened by clicking ‘Legend Setting’ button .



This ‘Legend Set’ dialog is similar with the one described in map data section. But here when the legend setting changed and applied, the contour or shaded shapes will be calculated over again. And the legend type can not be changed here. The legend setting could be edited manually. And a ‘Legend Breaks’ automatically setting tool was applied also. Open it with ‘New Legend or Color’ button.

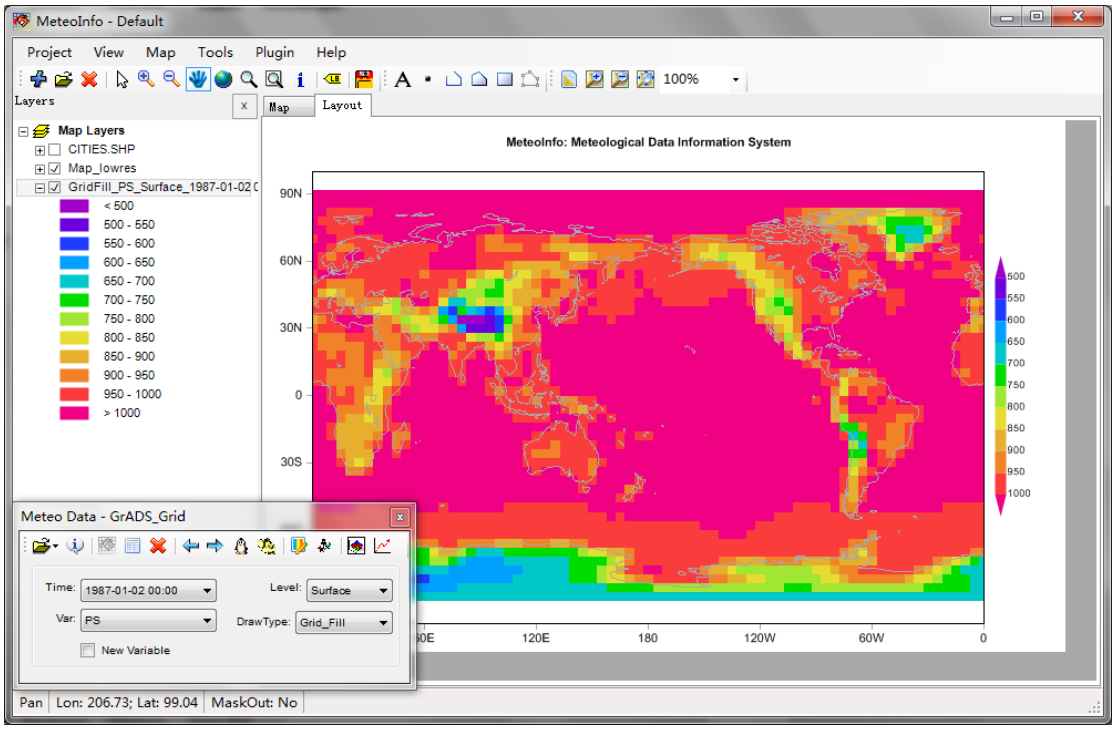


Change the interval to 4 and the color set to 'Shade Color' (the colors will be created from start color to end color). Then click 'New Legend' button. And then press 'Apply' button in 'Legend Set' dialog. The changed only could be saved after you click 'OK' button.

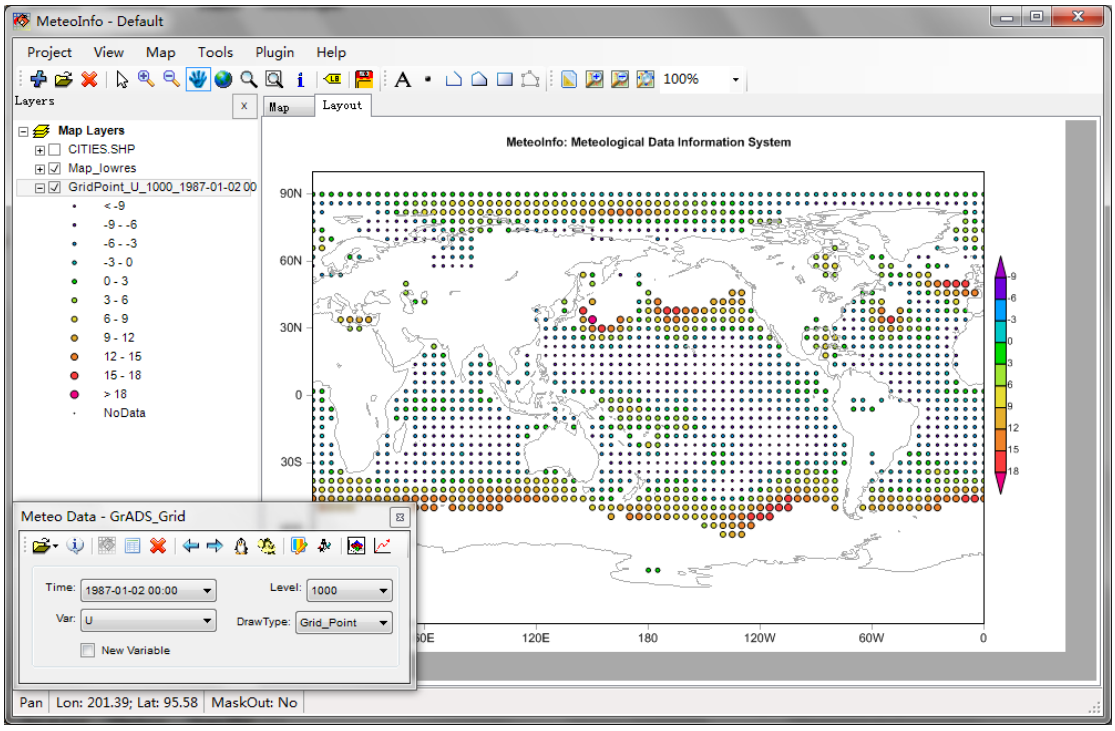


Grid Fill and Grid Point Graphic

With Grid Fill graphic, all grid data were plotted with rectangle polygons.

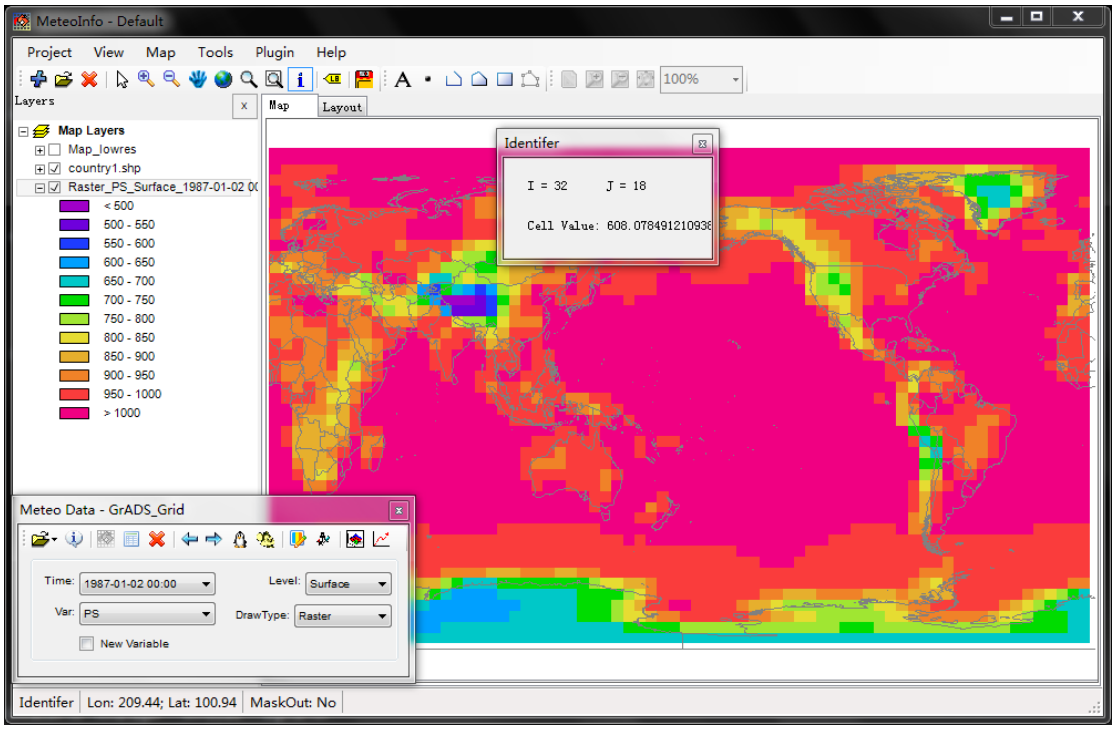


With Grid Point graphic, all grid points will be plotted as point shapes with grouped colors and sizes.



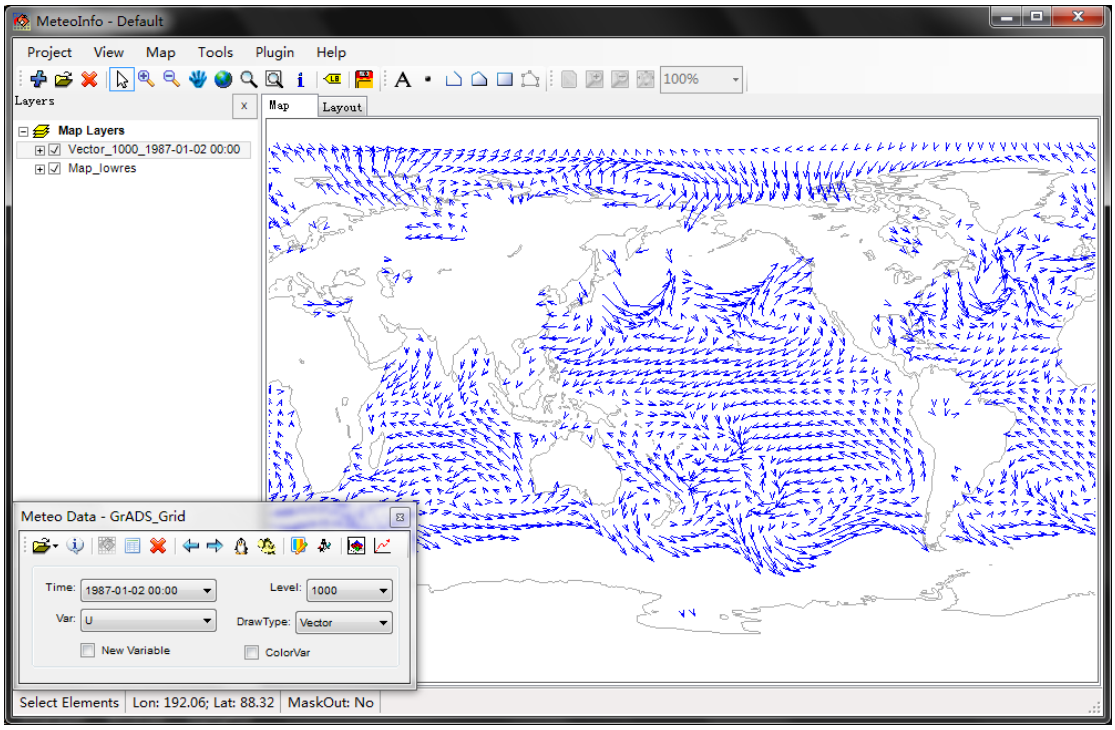
Raster Graphics

Raster option is much faster than the others for a grid data with a large number of cells. The data value of each cell can be identified by ‘Identifer’ tool.

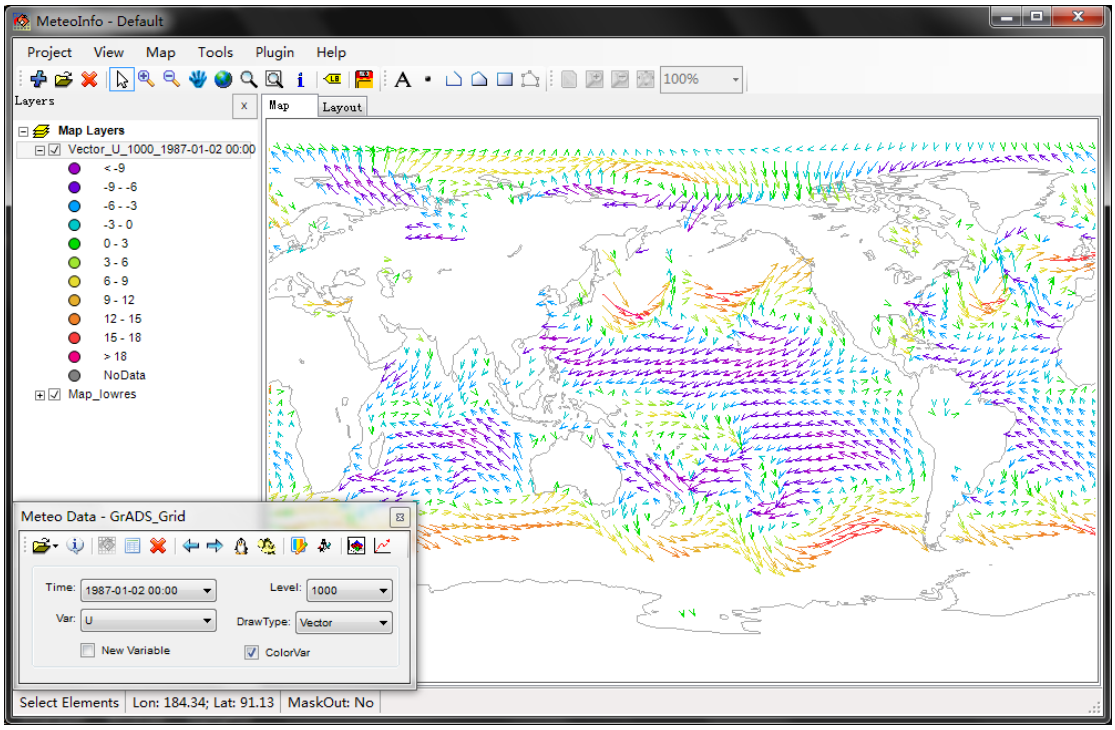


Vector Graphics

The wind field will be plotted according U and V weights.

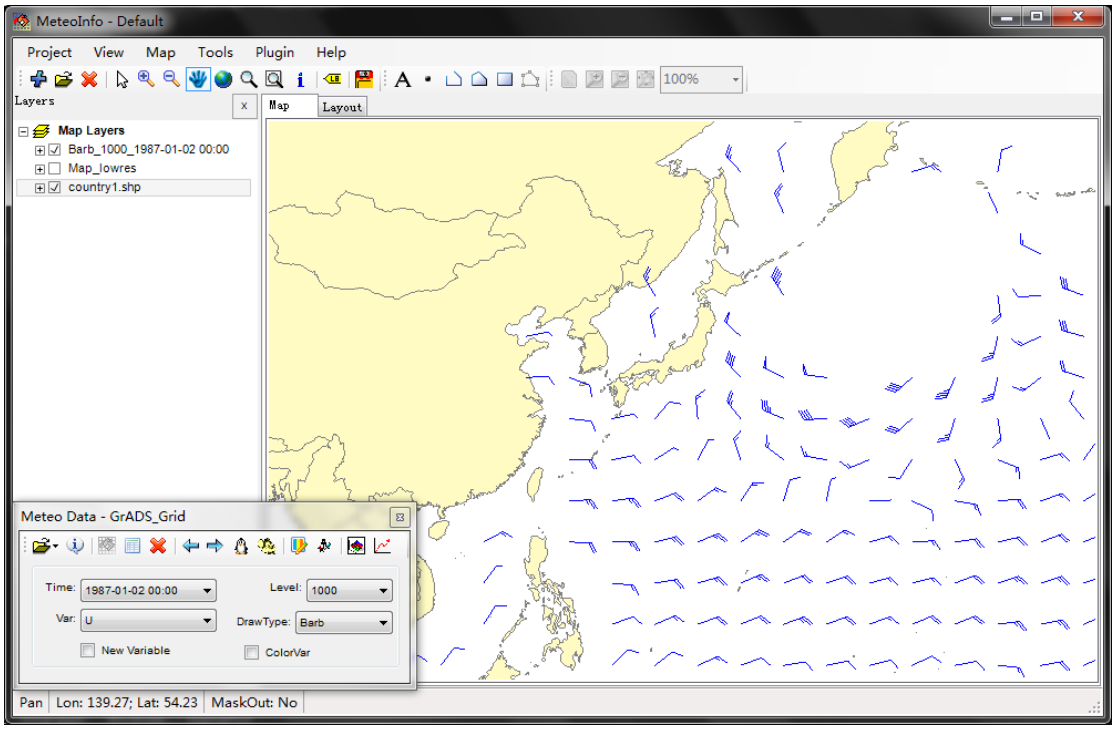


After select ‘ColorVar’ option, the wind symbols could be colored by third various selected in ‘Var’ item.

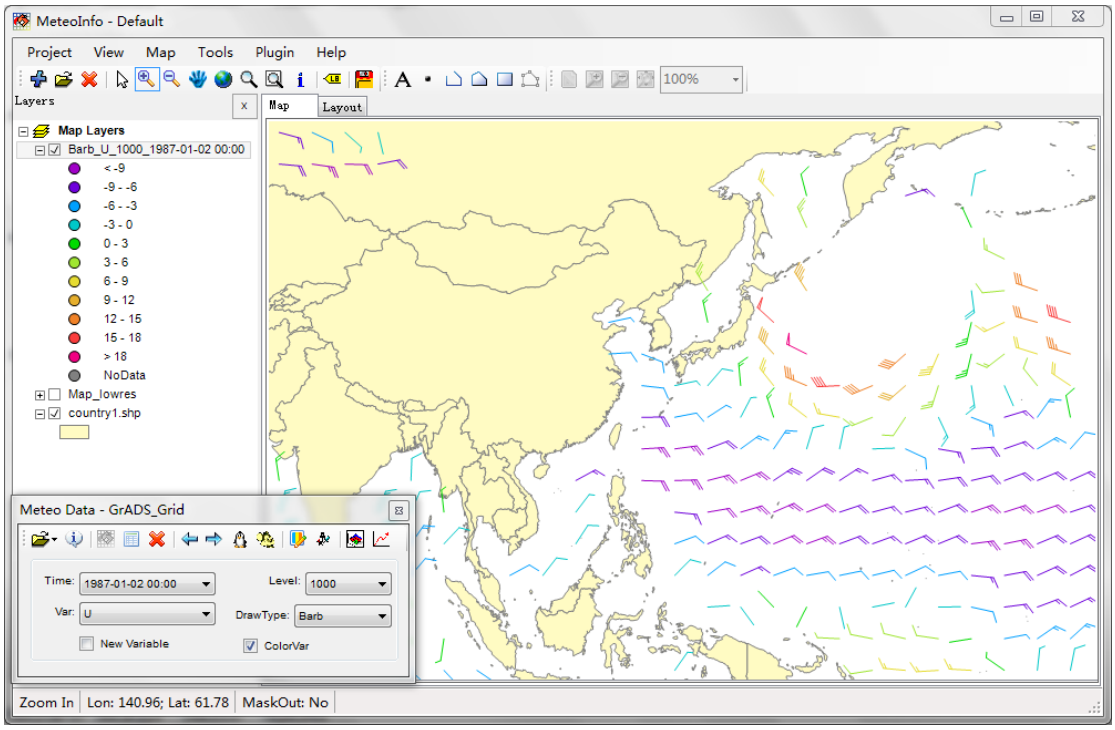


Barb Graphics

The wind field will be plotted as barbs.

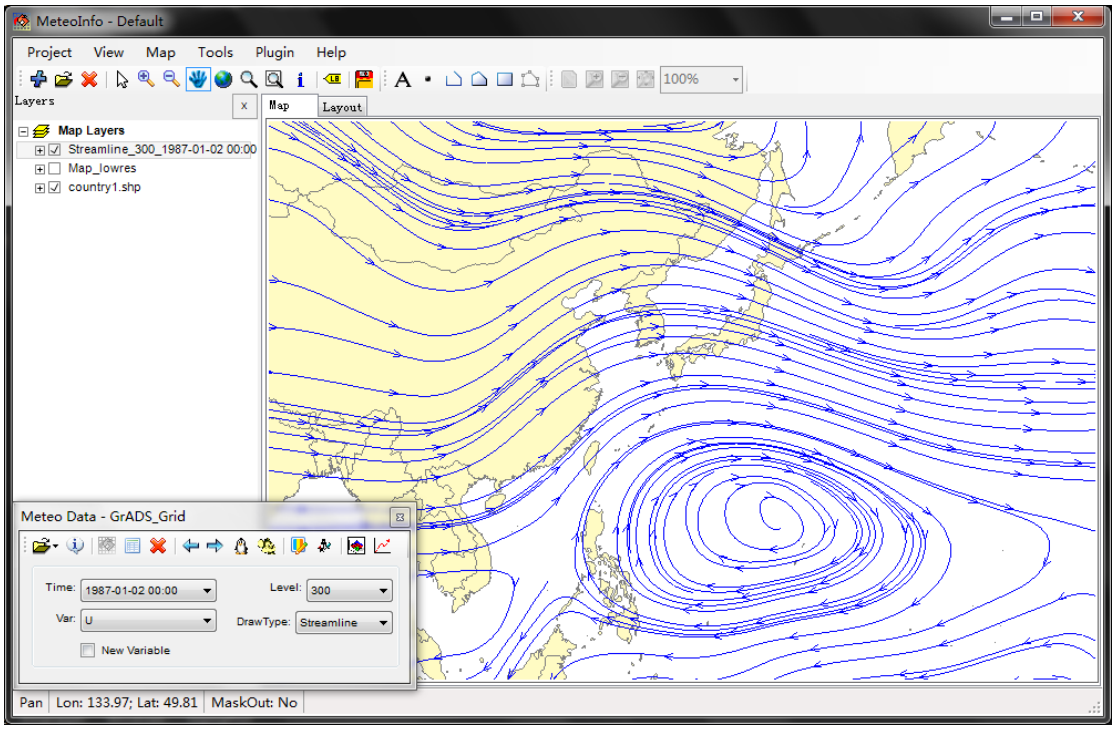


After select ‘ColorVar’ option, the wind barbs could be colored by third various selected in ‘Var’ item.

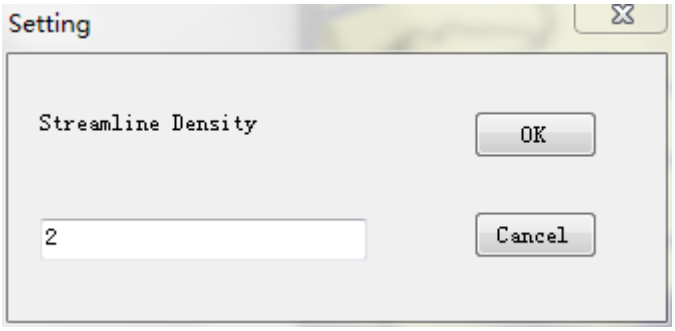


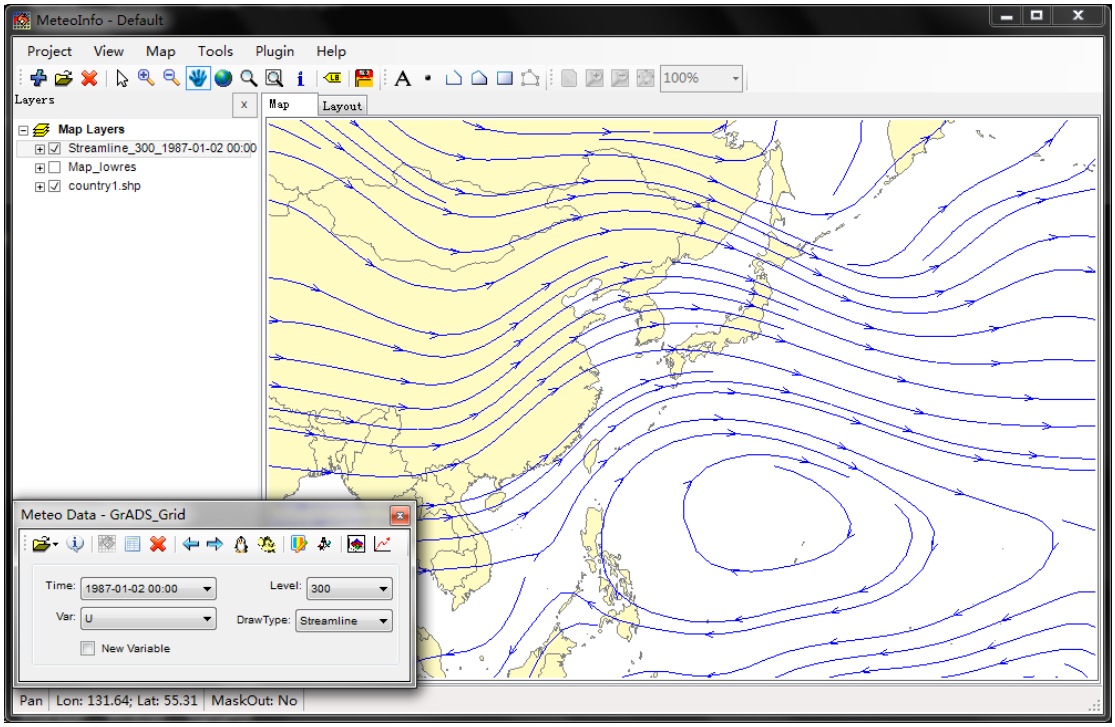
Streamline Graphic

The streamlines will be traced and viewed according to the U/V components of the wind.



The streamline density could be adjusted from 1 to 10. The default value is 4. If change the value to 2, the streamlines look like following.





Plot Station Data

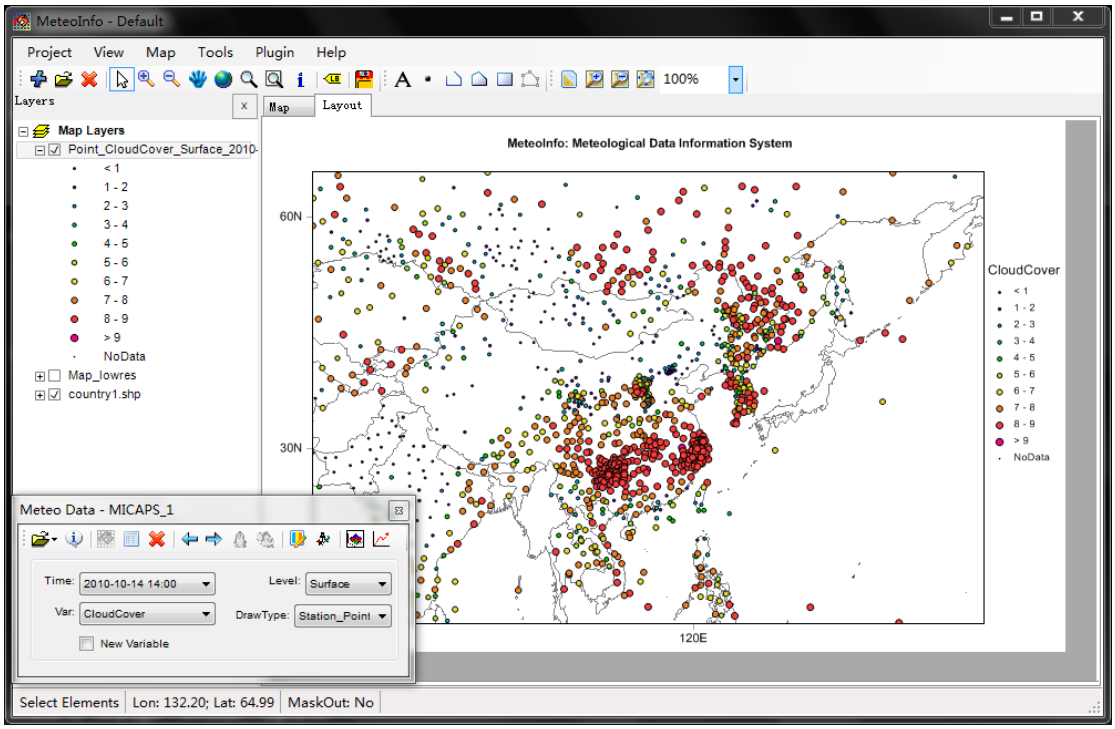
Station data could be plot as following graphic types.

- ✧ Station Point
- ✧ Contour
- ✧ Shaded
- ✧ Barb
- ✧ Weather Symbol
- ✧ Station Model
- ✧ Station Info

The following pictures were from MICAPS 1 data ‘10101414.000’.

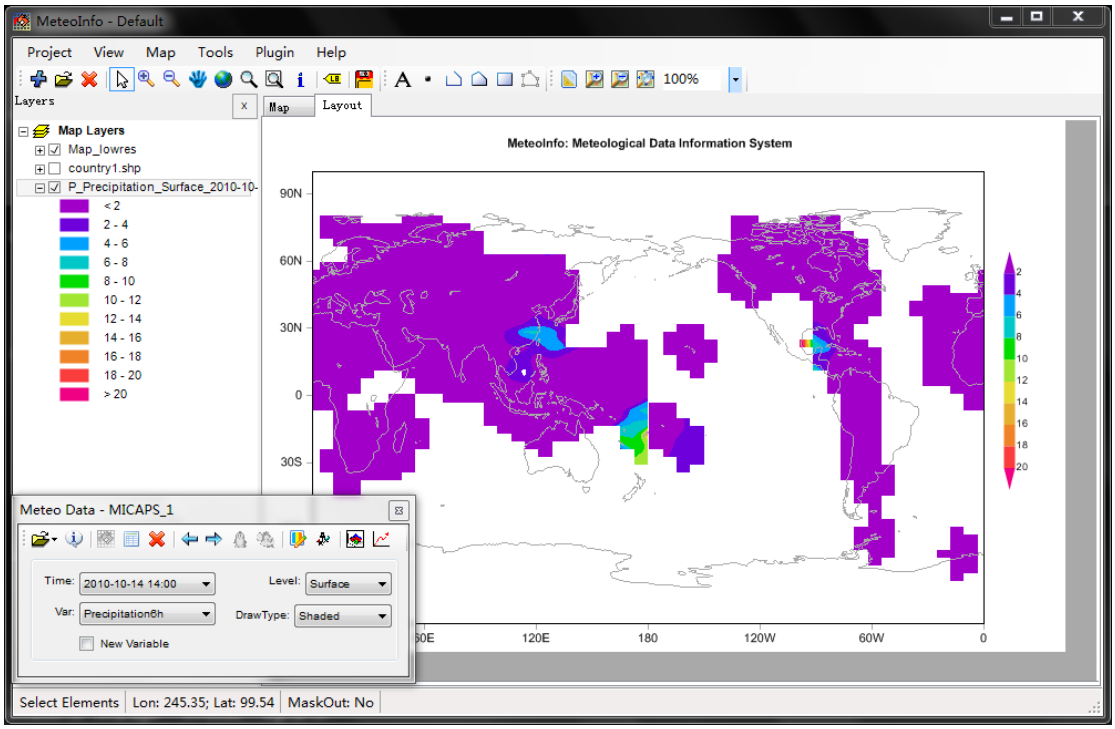
Station Point Graphic

Set ‘Draw Type’ to ‘Station_Point’.



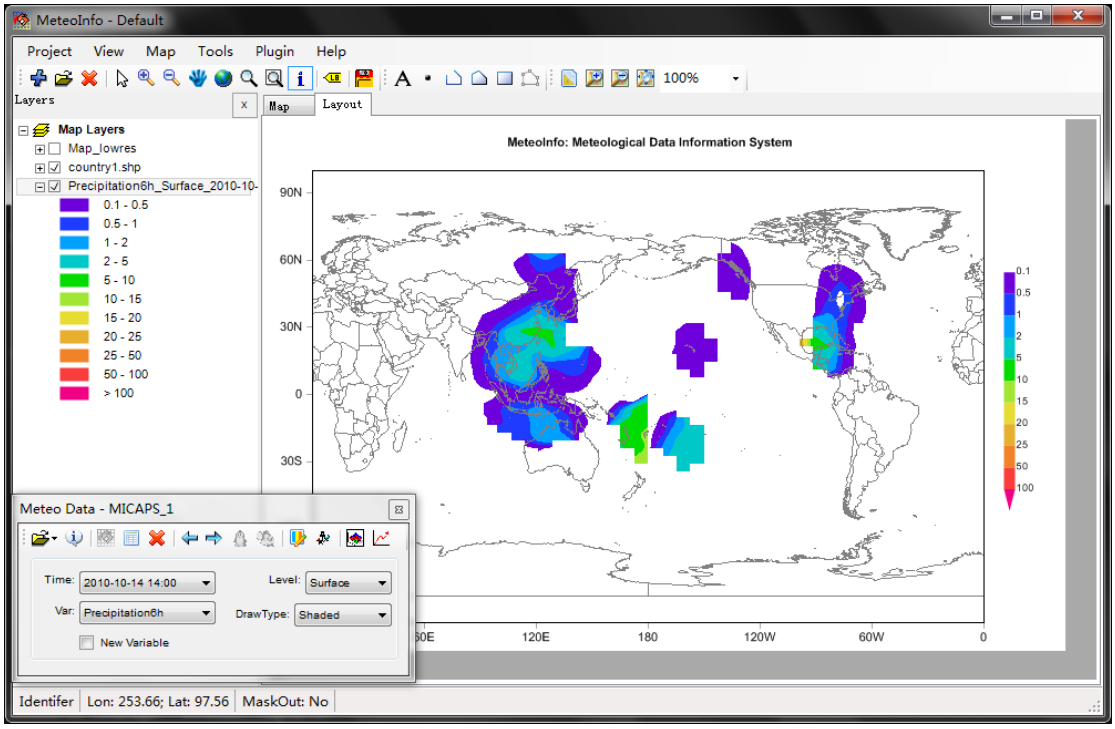
Contour and Shaded Graphics

To draw the contour or shaded graphics, the station data have to be interpolated to grid data. The software will set the interpolation parameters automatically. The automatic setting is not good in this case, so we should change the setting.

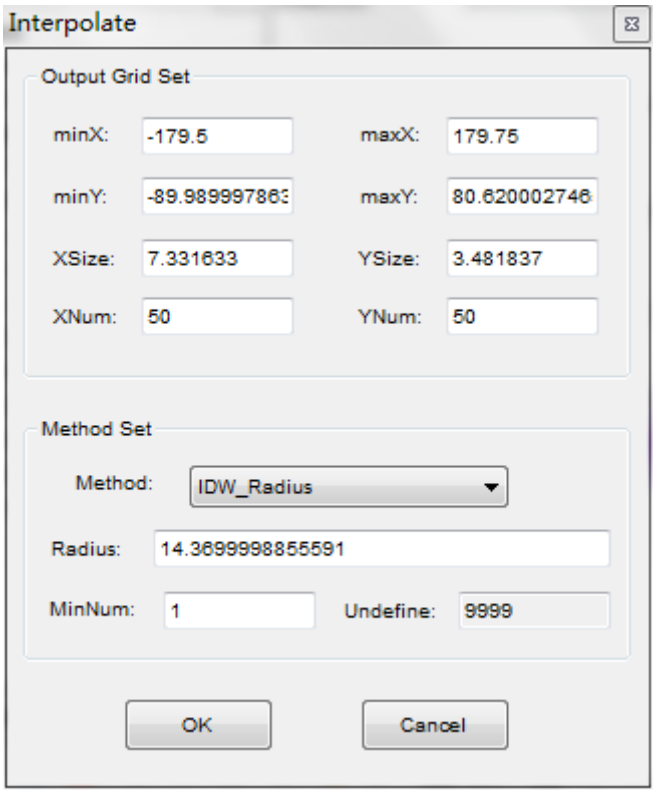


After legend was reset, the graphic looks as following picture. It is much better, but maybe

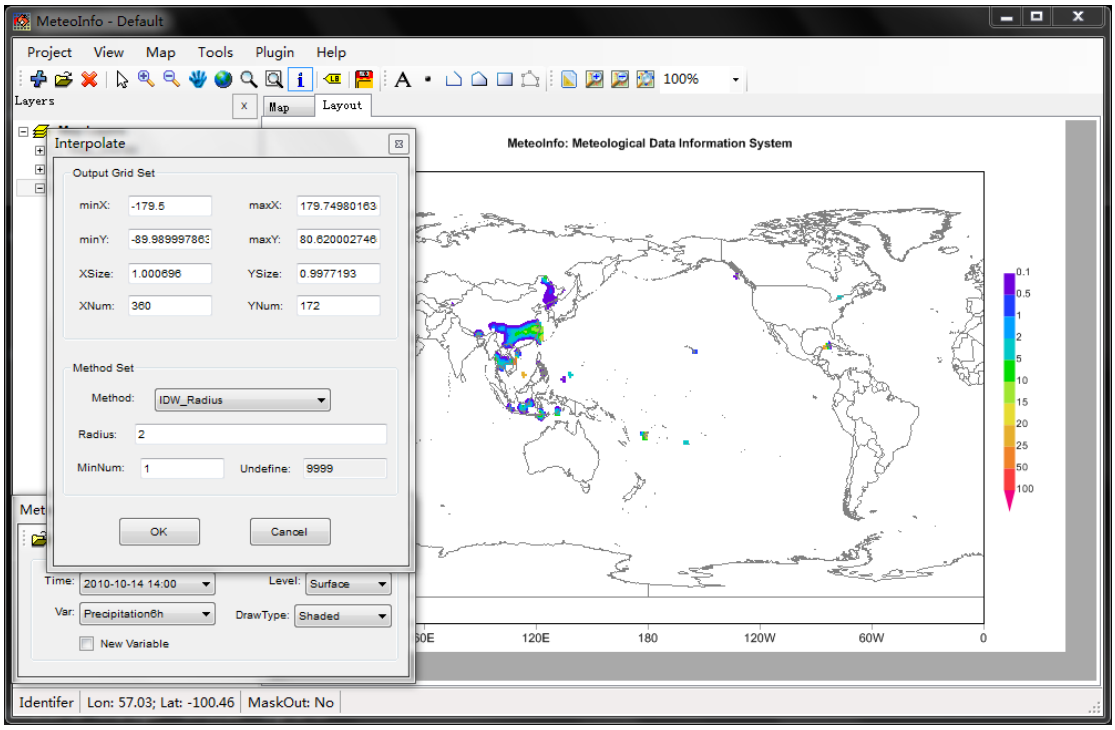
you still don’t like it. Then you could change the interpolation setting to see what will happen.



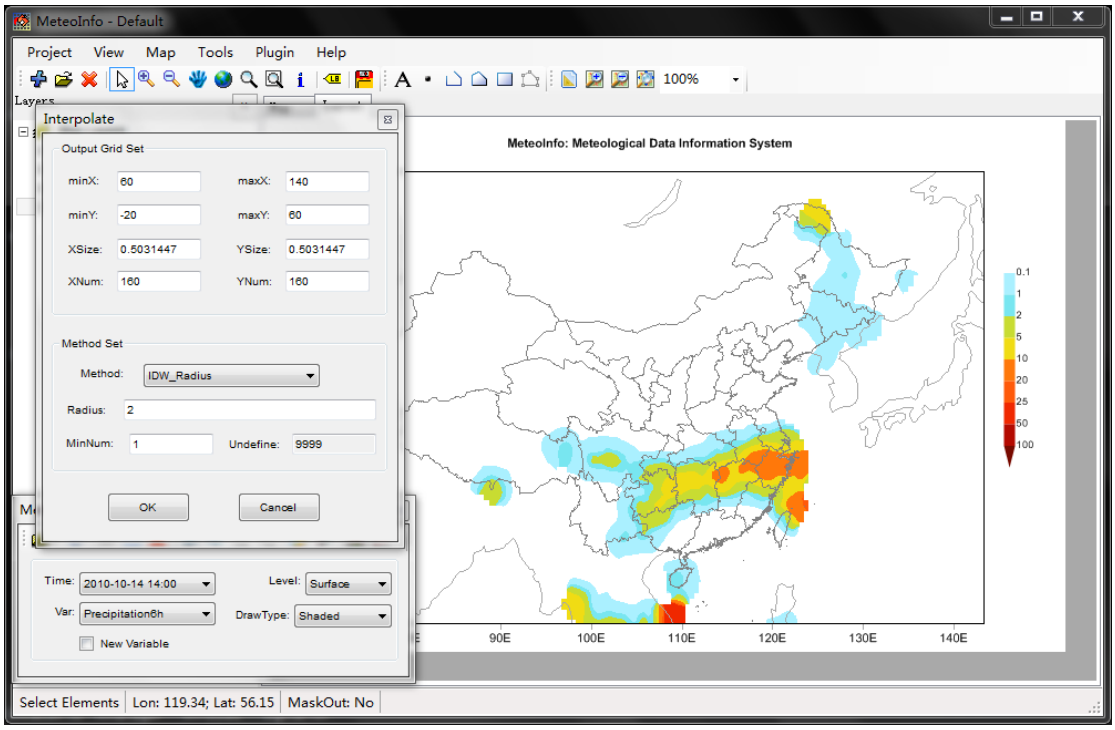
Click ‘Setting’ button in ‘Meteorological Data’ dialog to open ‘Interpolate’ dialog. In which output grid and interpolation method can be set. IDW (Inverse Distance Weighted) and Cressman analysis methods are applied at present. IDW includes IDW_Radius and IDW_Neighbors options. With IDW_Radius, the grid point without station point found in radius will be set as undefined (which will not be used in contour tracing process). With IDW_Neighbors option, all grid points will have values calculated from the nearest station points. The continuous distributed variables such as temperature may suite for IDW_Neighbors option. In this case, we first select interpolation method as IDW_Radius.



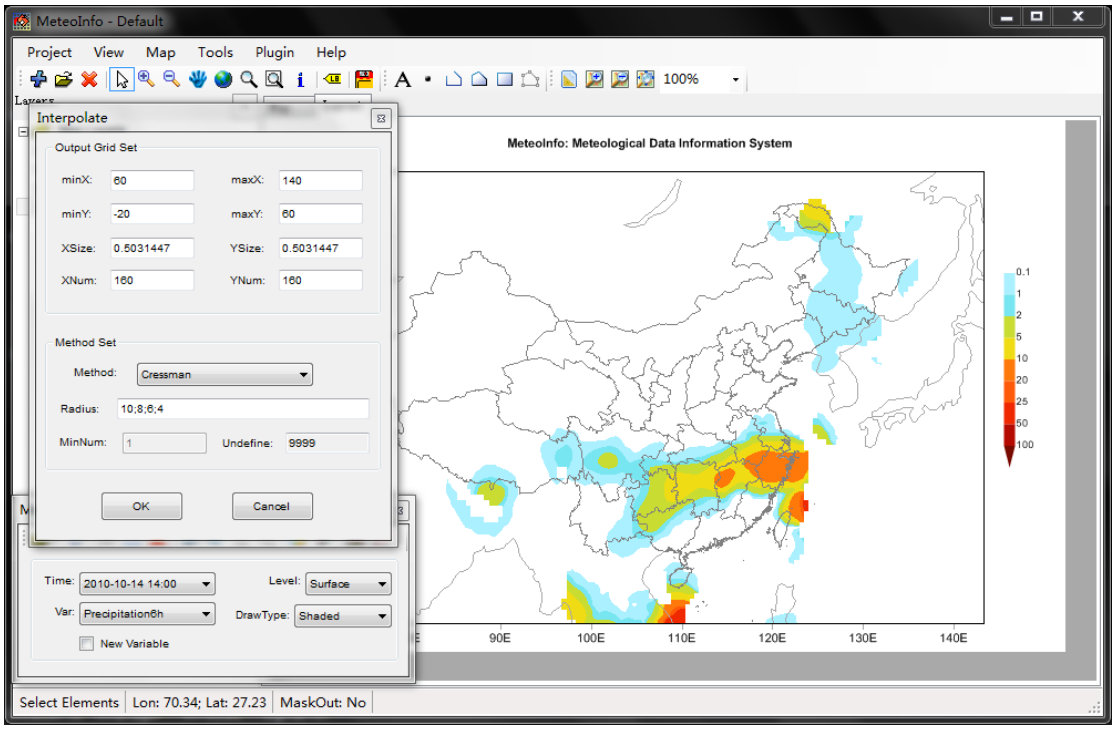
After you changed the setting, press ‘OK’ button to save the setting and the ‘Interpolate’ dialog will be closed also. Then press ‘Draw Data’ button in ‘Meteorological Data’ to redraw the figure with new setting. If you set ‘XSize’ and ‘YSize’ to 1, and ‘Radius’ to 2, you could get following picture. With small ‘XSize’ and ‘YSize’ means much more grid points have to be interpolated, so much more time will be cost for it.



If you just want to see the data in a certain area, such as China, you could change the output grid extent.

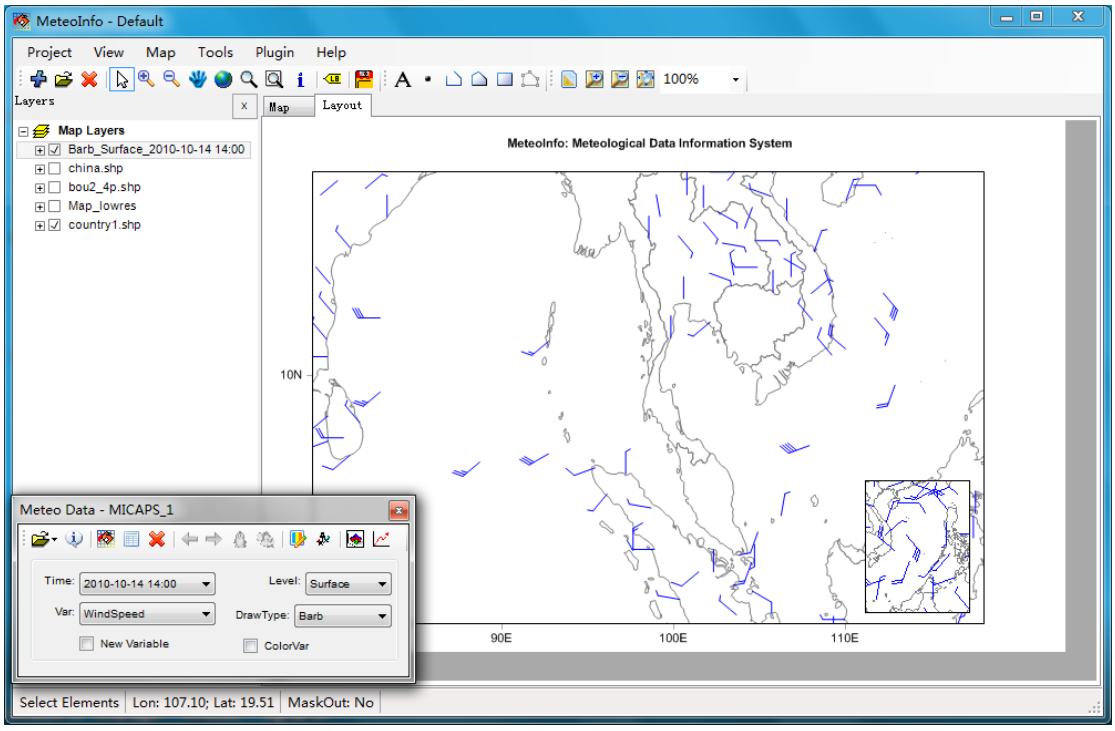


The Cressman objective analysis scheme (Cressman, 1959) is known as the successive correction method. It achieves its result by forcing convergence of the data to observed, interpolated values using multiple iterations. The example Cressman analysis result is plotted below.



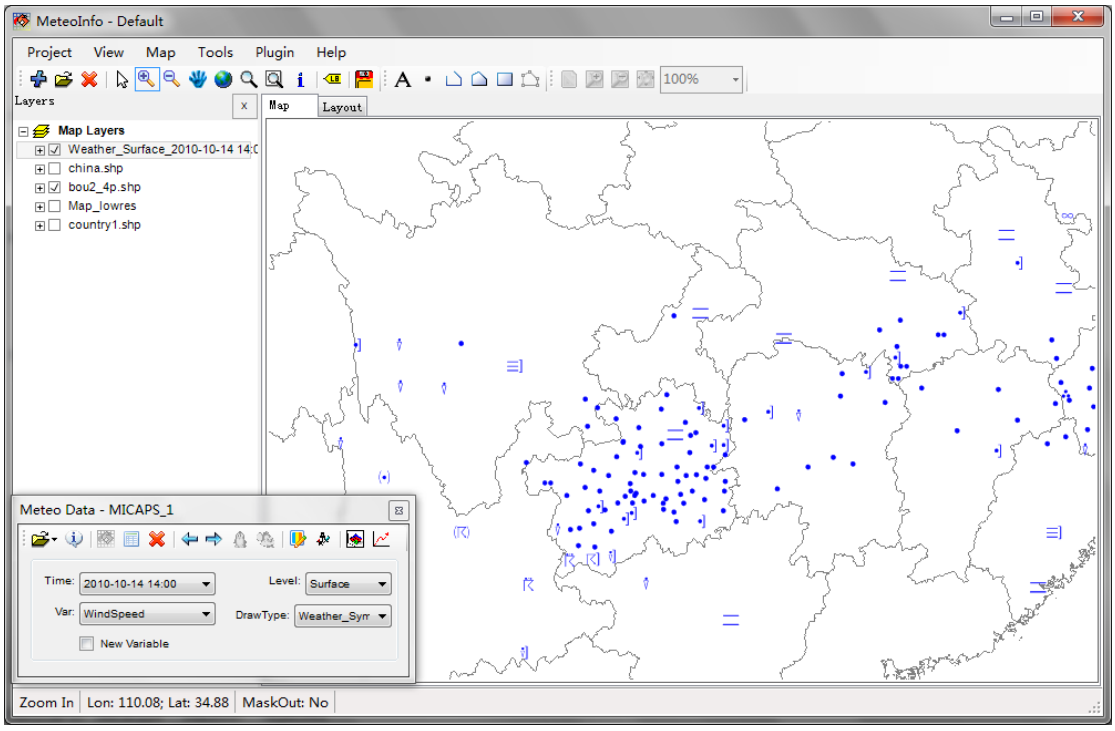
Barb Graphic

Select ‘Draw Type’ to ‘Barb’. Click ‘Draw Data’ button.

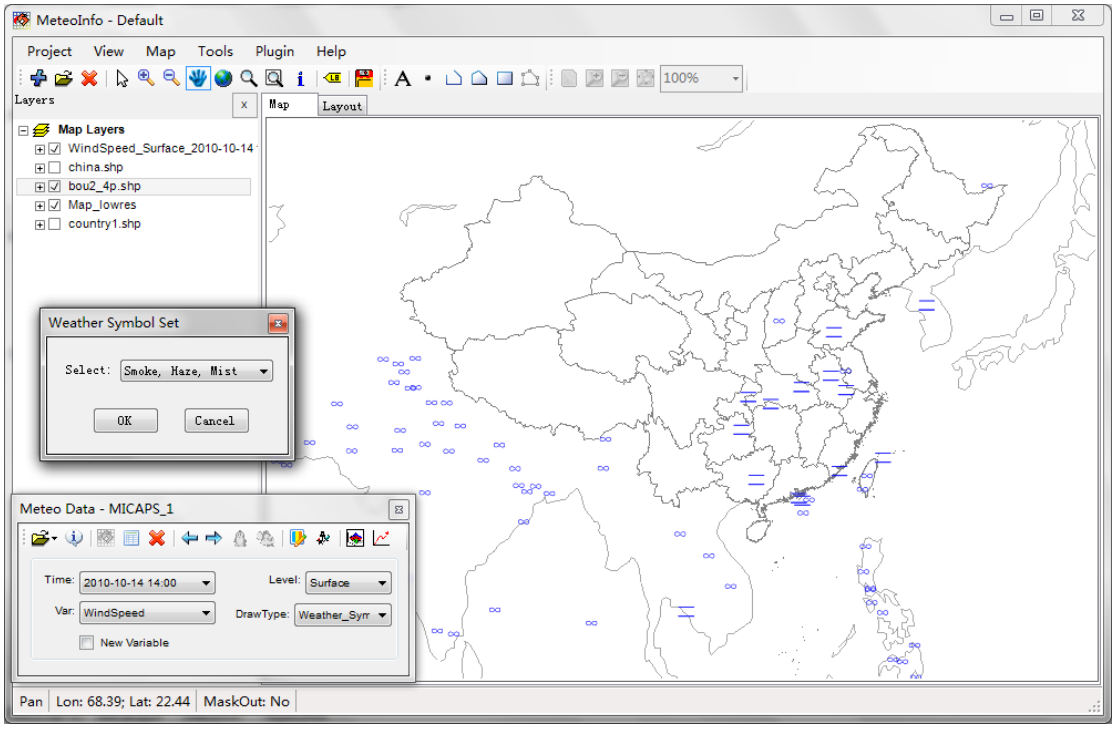


Weather Symbol Graphic

With default set, all kinds of weather were showed in the Map.

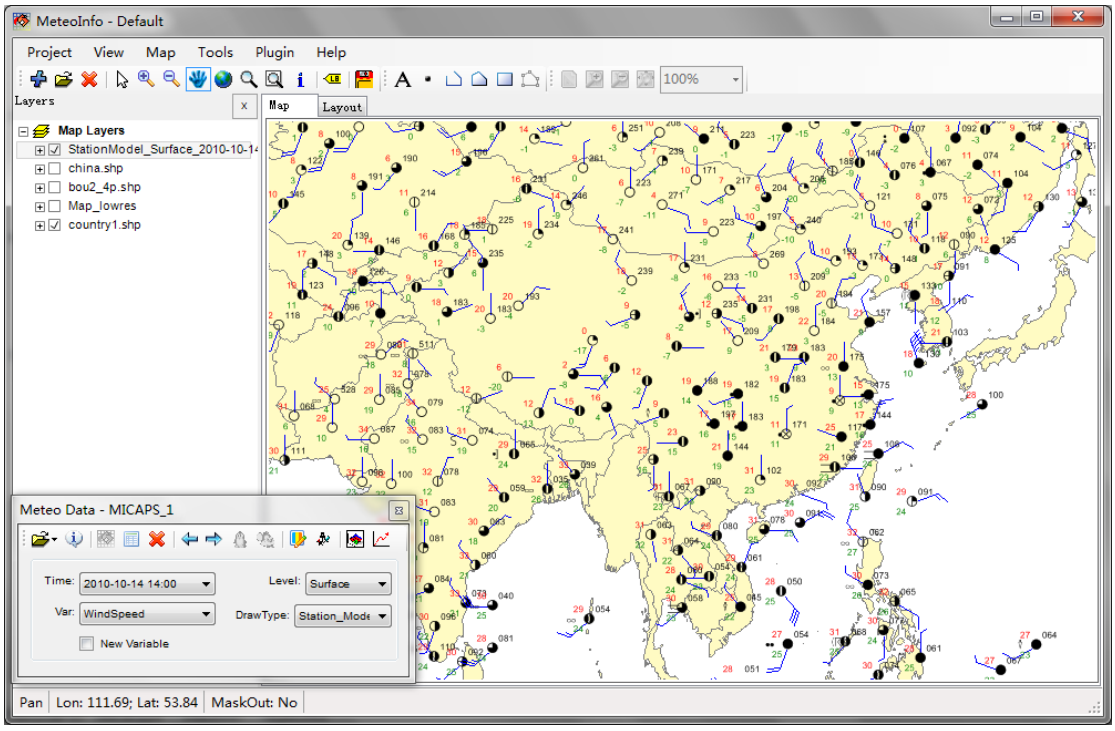


Click ‘Setting’ button and select one weather group in ‘Weather Symbol Set’ dialog, and then press ‘OK’ button.



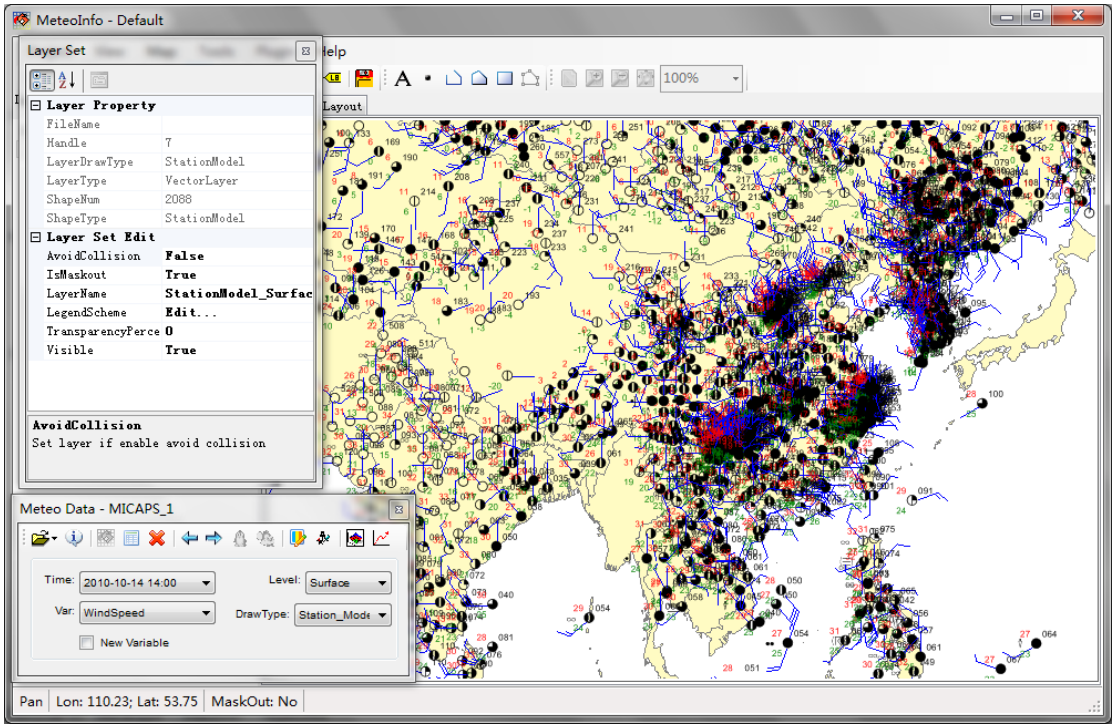
Station Model Graphic

Open a station data set (METAR or MICAPS1 format), and select ‘DrawType’ as ‘Station_Model’. After press ‘Draw Data’ button, the station model graphic is presented.



The ‘AvoidCollision’ property was set to ‘True’ automatically when the station model layer was created to avoid the overlap of the station model symbols. When the map was zoomed in or zoomed out, more or less detailed stations were plotted automatically.

If ‘AvoidCollision’ is set to ‘False’, all stations in the map extent will be plotted.

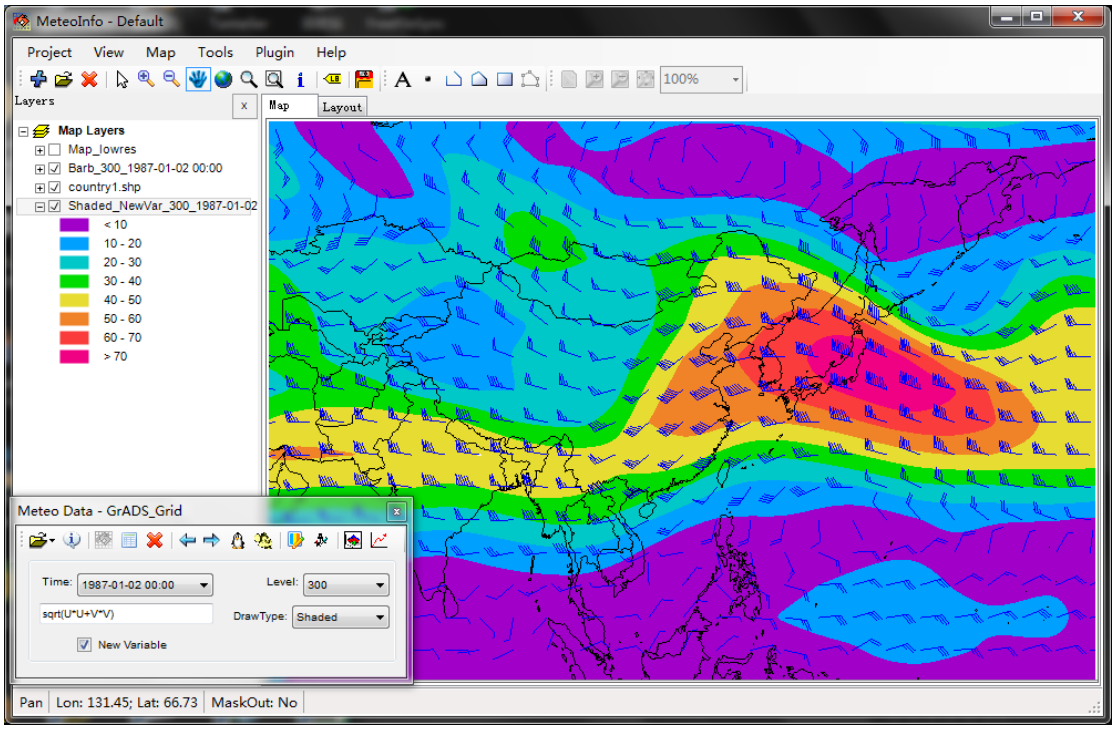


Plot new variable data

A simple math parser function is included, so user can get a new variable by inputting a formula including existing variables. For example, a new wind speed variable can be created from existing ‘U’ and ‘V variables by the formula ‘sqrt(U*U+V*V)’.

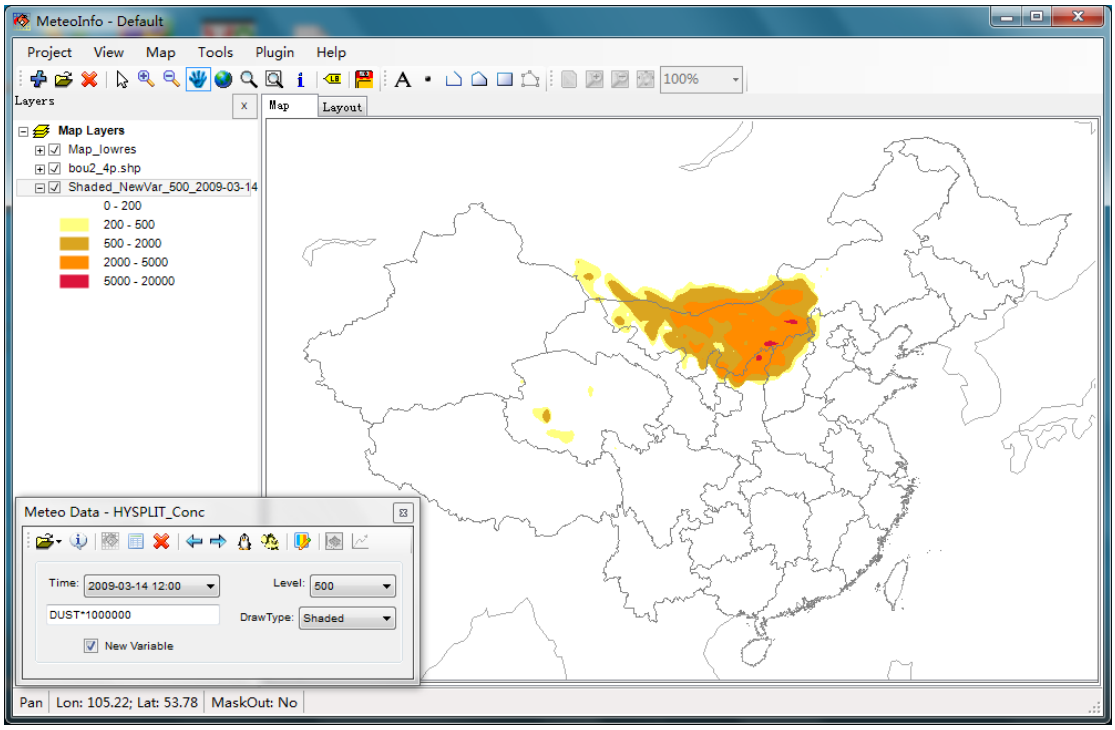
Supported operators are: ‘+’, ‘-’, ‘*’, ‘/’, ‘%’, ‘^’.

Supported functions are: ‘abs’, ‘acos’, ‘asin’, ‘atan’, ‘cos’, ‘exp’, ‘log’, ‘log10’, ‘sin’, ‘sqrt’, ‘tan’.

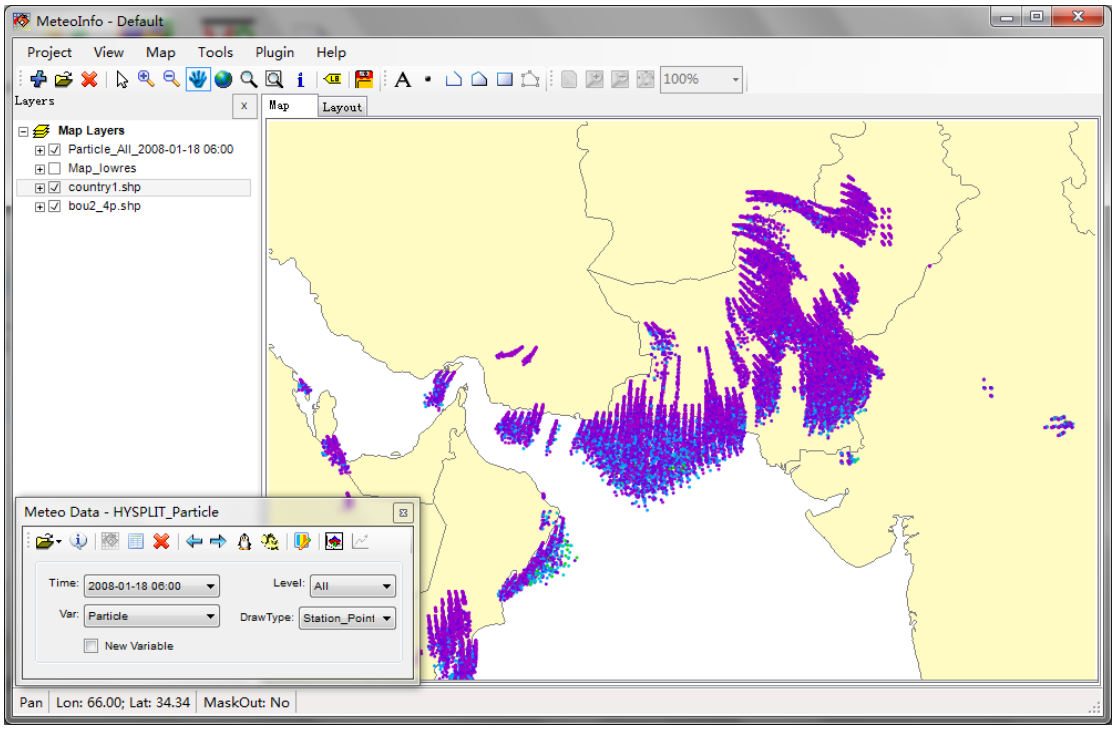


Plot HYSPLIT Data

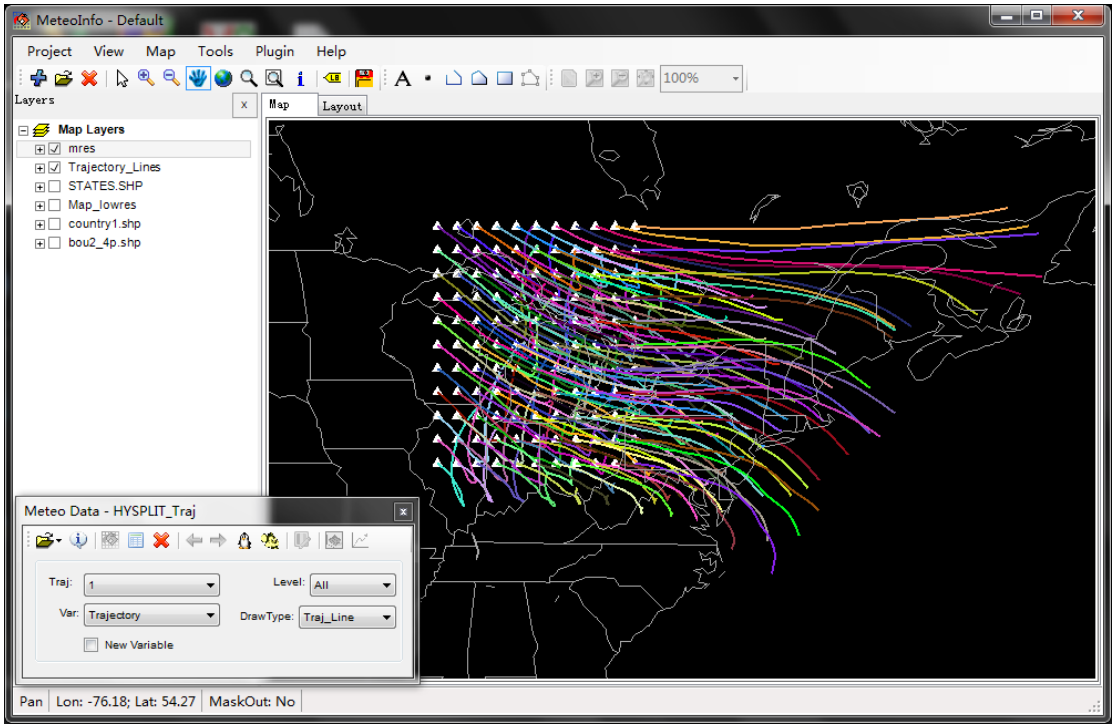
Concentration data plot.



Particle data plot.



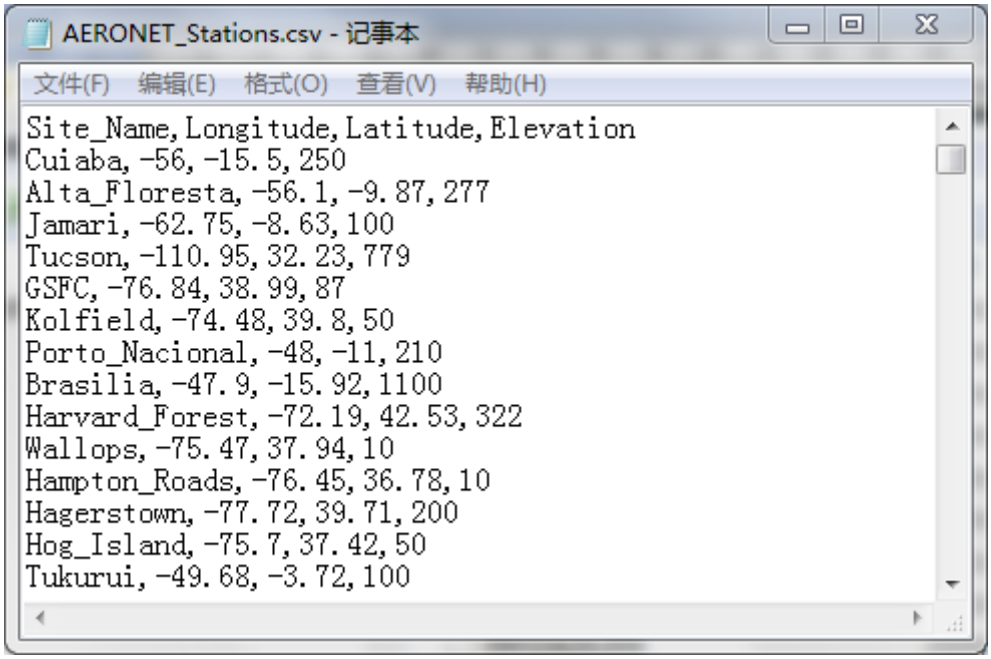
Trajectory data plot.



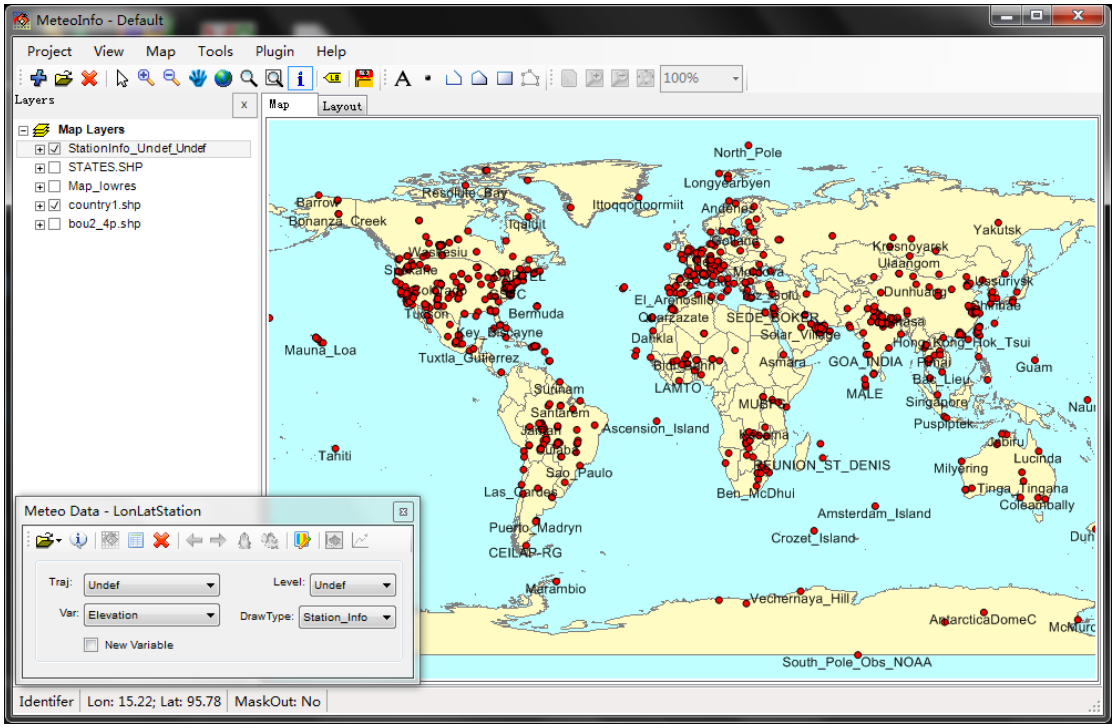
Lon/Lat Station Data

The station ASCII data with longitude and latitude columns could be plotted in MeteoInfo. The data should be comma-delimited. The file must contain column titles as the first row. The first column must be station identifier (name or code), and the second column must be longitude, and

the third column must be latitude. The file could contain many columns.

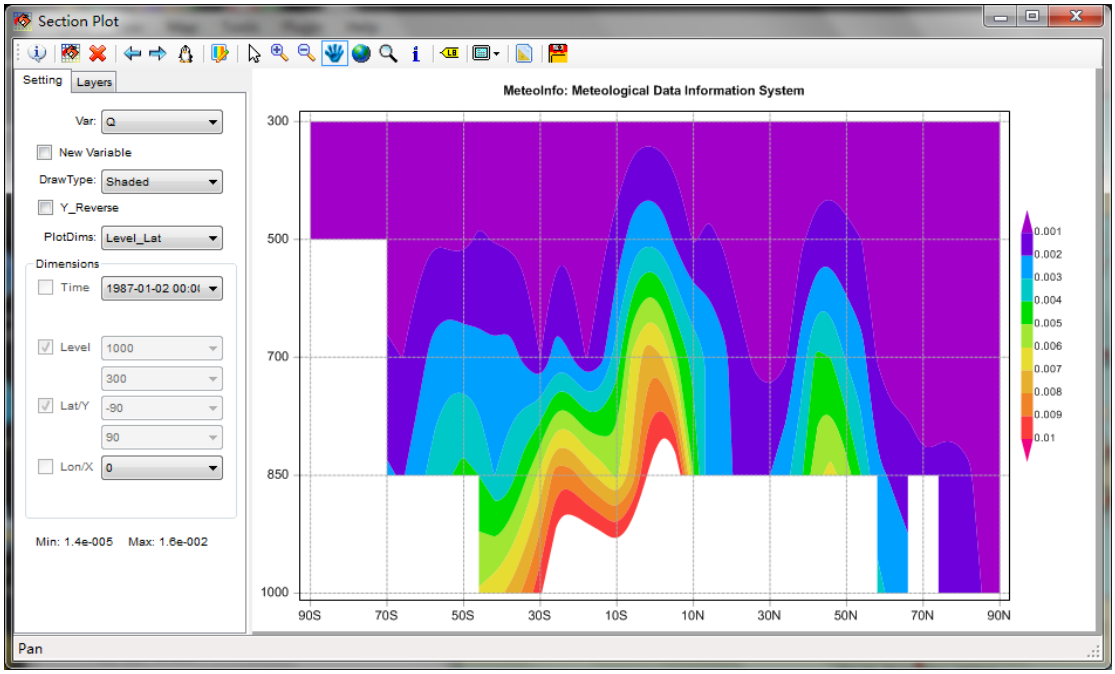


Open the data file with ‘Lon/Lat Stations’ menu, and select ‘Draw Type’ to ‘Station_Info’. Then click ‘Draw Data’ button to create a station layer with all columns as attribute data.

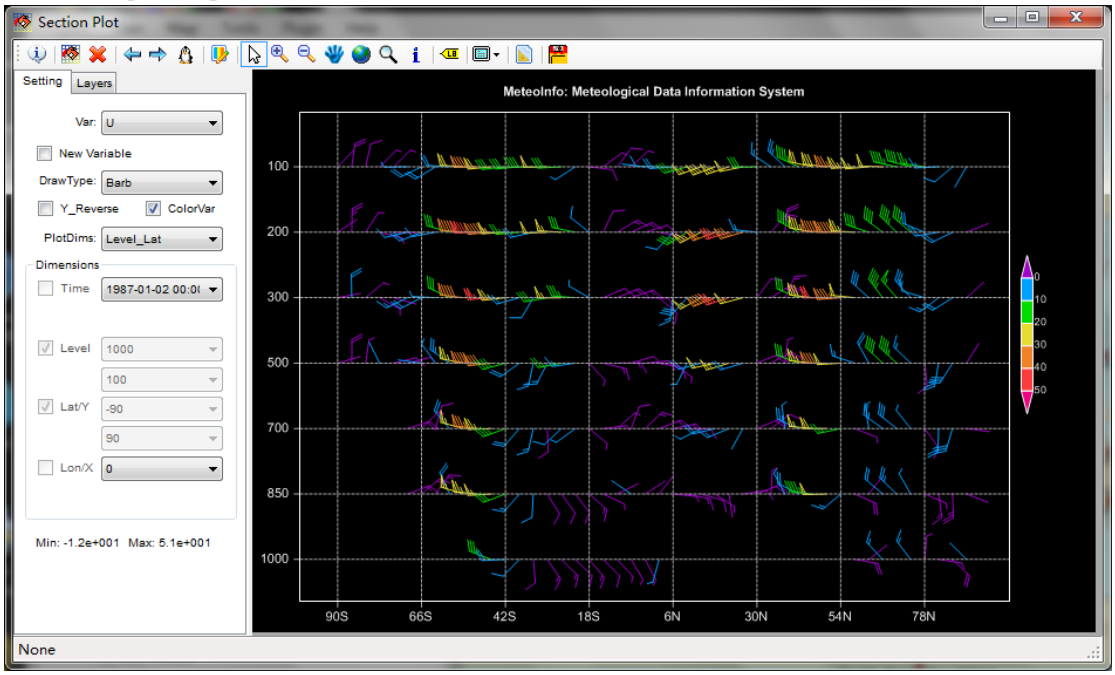


Section Plot

Click ‘Section Plot’ button to open ‘Section Plot’ window. Select ‘Var’, ‘DrawType’ and ‘PlotDims’ and set the dimensions and then press ‘Draw Data’ button, section plot will be plotted. For this kind of plot, there are two variable dimensions.

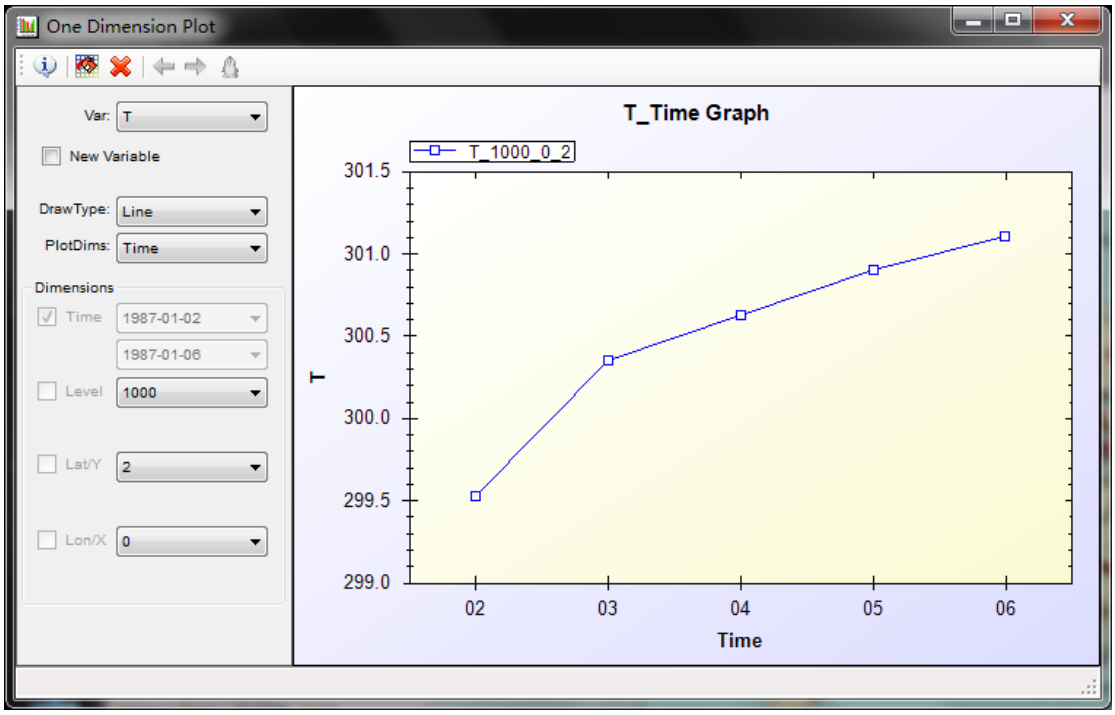


Wind profile plot.



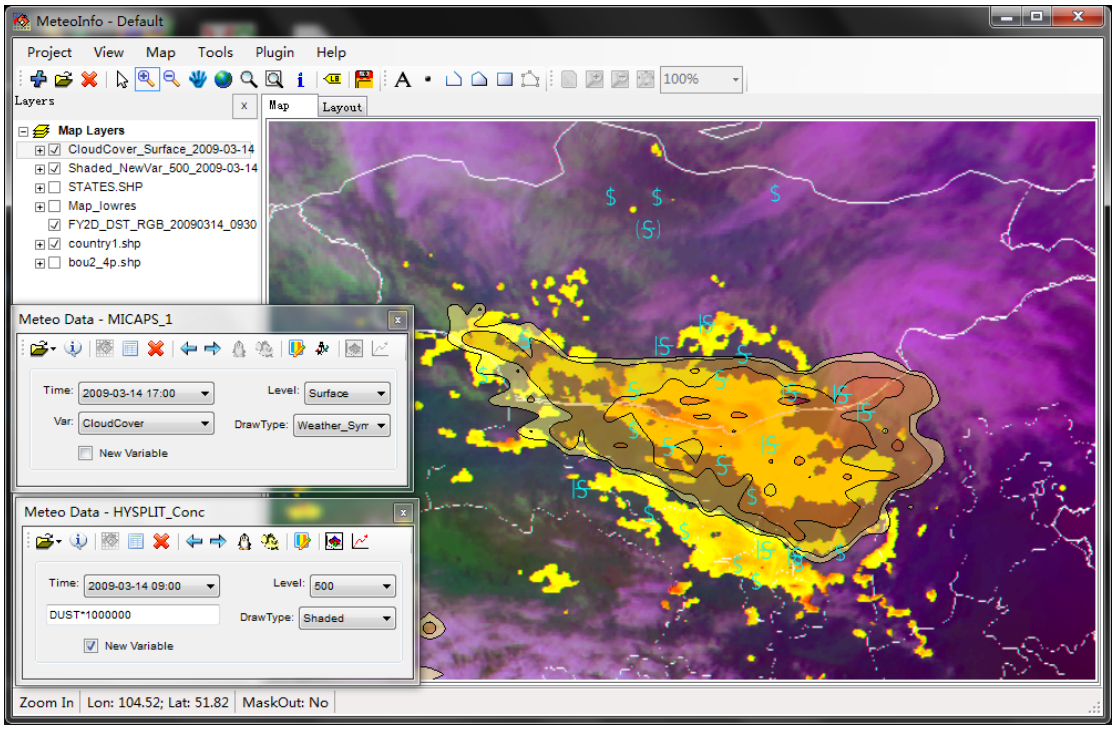
One Dimension Plot

Click ‘One Dimension Plot’ button to open the window. The setting is similar with ‘Section Plot’, but only one dimension is variable.



Model Verification Example

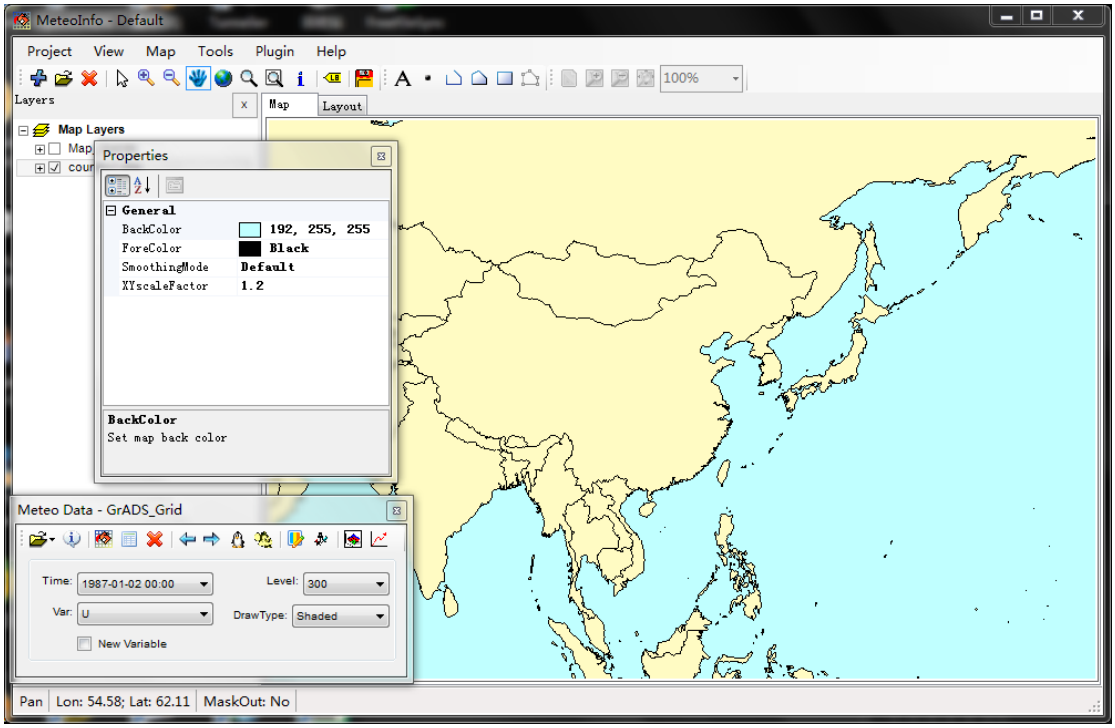
With abilities of view model result, station observation and satellite image of MeteoInfo, it is quit easy to compare the model result with observations. Following is a case of dust storm forecast verification using station weather data and satellite data. Polygon shapes could be set as transparency in MeteoInfo.



Map and Layout Set

Map Property

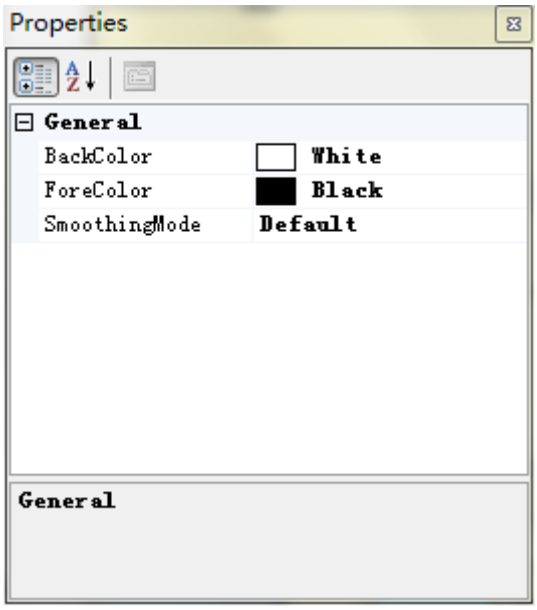
Click ‘Map Property’ sub-menu item under ‘Map’ menu item to open the properties dialog. The property of back color, fore color, smoothing mode and x/y scale factor could be changed.



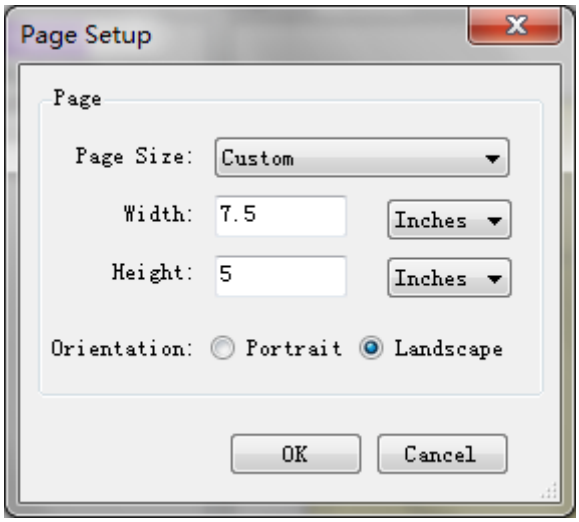
GDI+ graphic technology was used in MeteoInfo. The technology applied some smooth modes. With default smooth mode, the graphic could draw fast but with alias. Higher quality graphic could be got with ‘HighQuality’ or ‘AntiAlias’ smooth mode.

Layout Property

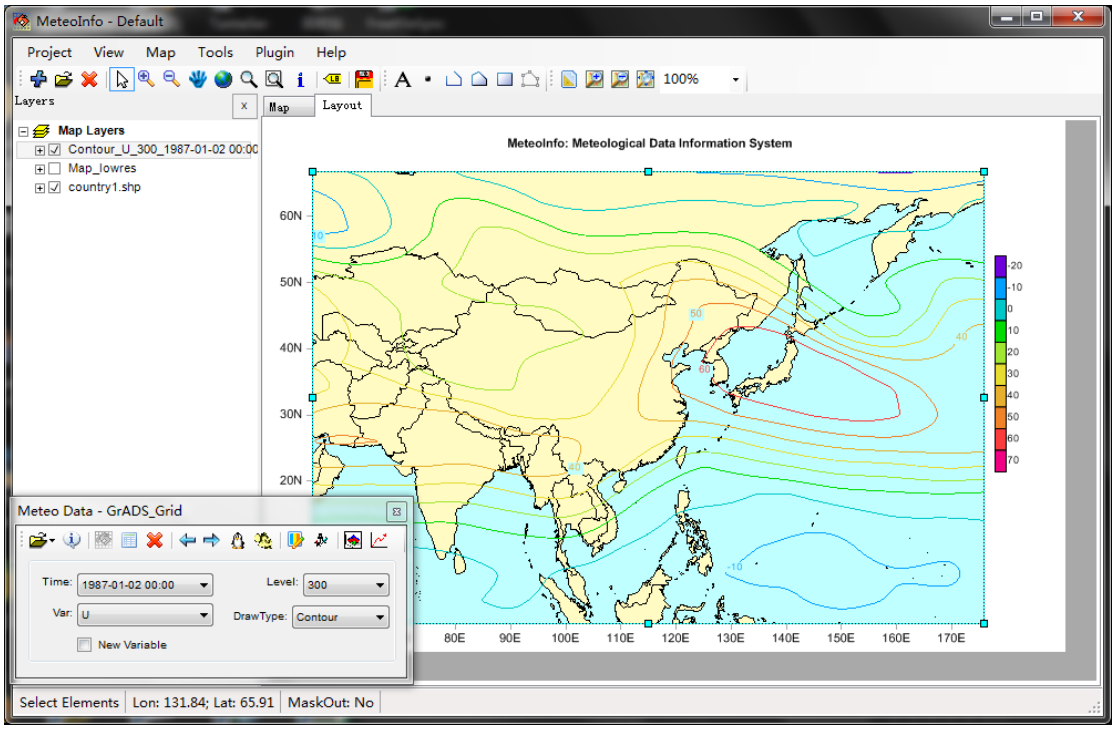
Click ‘Layout Property’ sub-menu item under ‘Map’ menu item to open the properties dialog. The property of back color, fore color and smoothing mode could be changed.



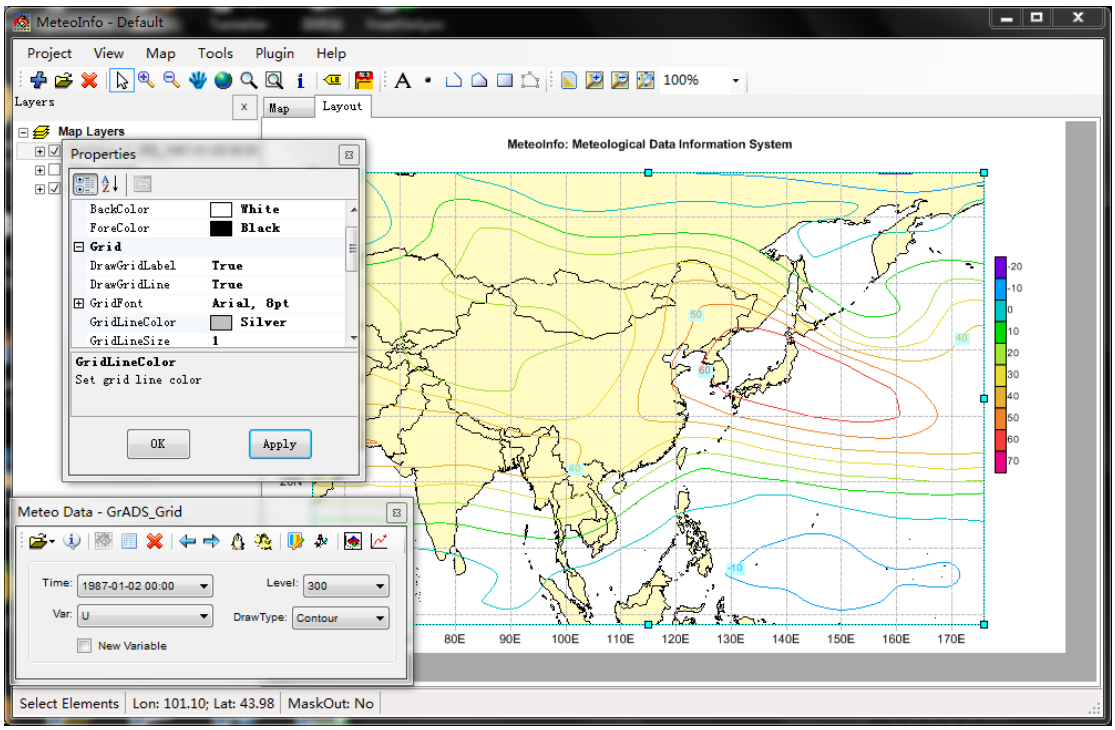
The tool strip of layout is enabled when the layout view is active. The layout page can be zoomed by the zoom buttons. The page size of the layout could be set by 'Page Set' button.



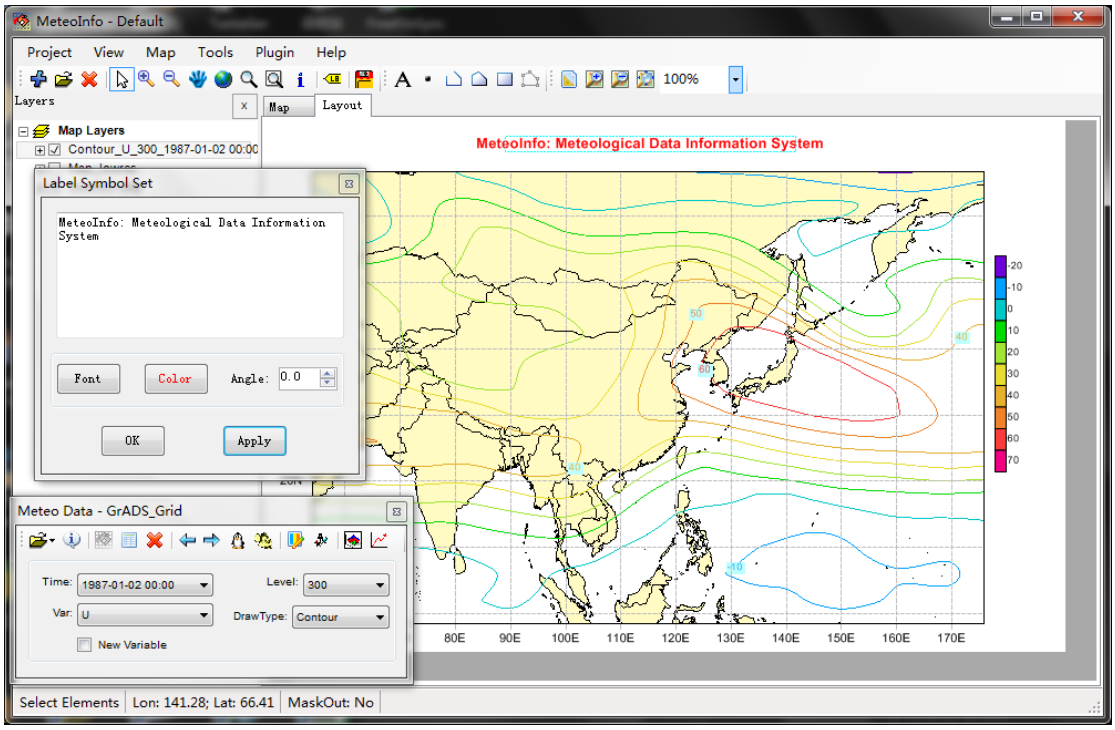
After select the 'Select Elements' tool button , the elements in the layout view could be selected and be moved or resized.



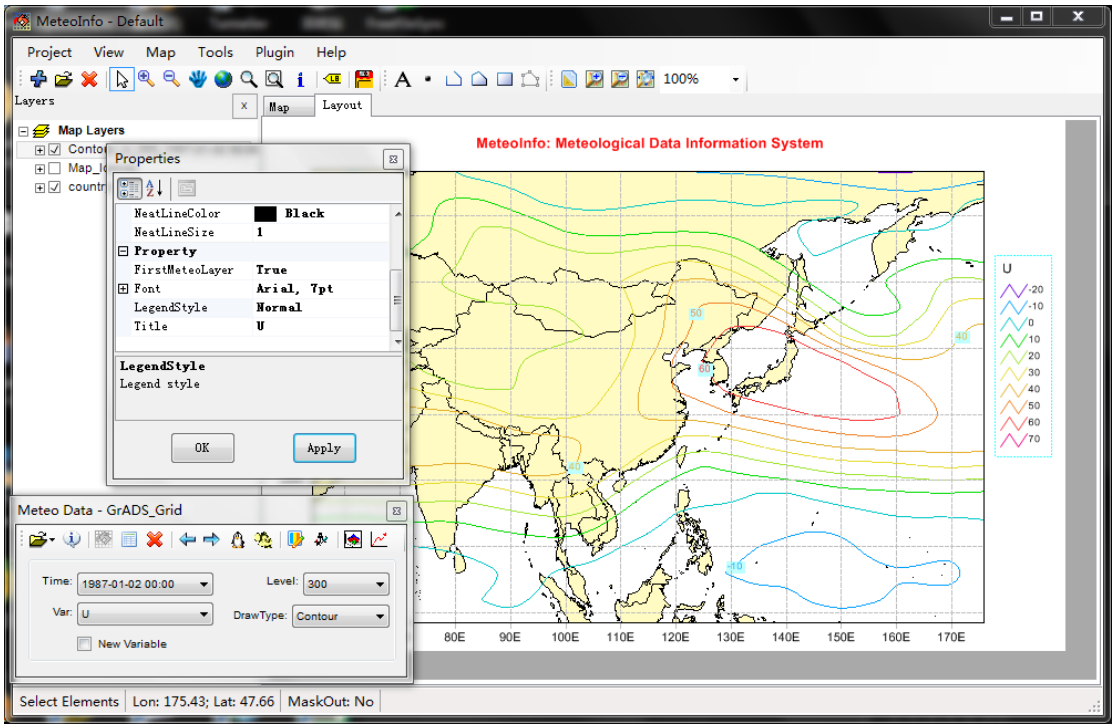
Double click the map element to open map property dialog and edit the properties.



Double click the title text to open text property dialog and edit the properties.

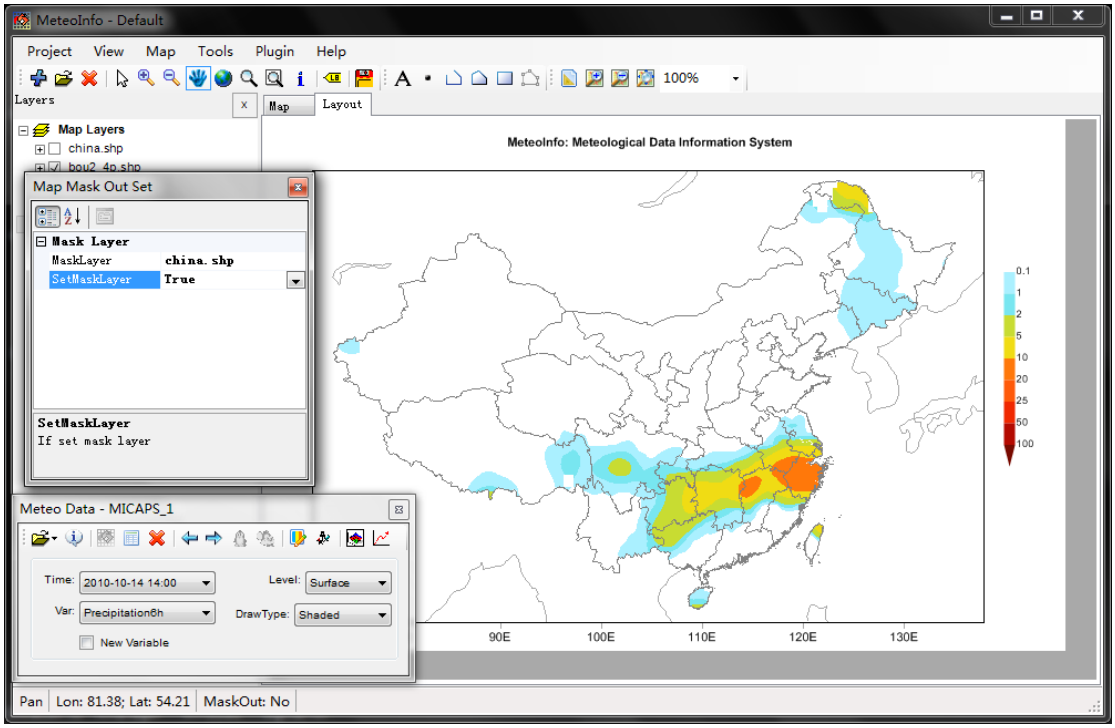


Double click the legend to open legend property dialog and edit the properties. The legend scheme of the first meteorological layer will be plotted when the 'FirstMeteoLayer' is 'True'. On the other hand, the legend scheme of the first expanded layer in layer management section will be plotted. 'SingleSymbol' legend scheme will not be plotted. There are three kinds of legend types: 'Bar_Vertical', 'Bar_Horizontal' and 'Normal'. The point, polyline or polygon symbol of each legend break is plotted with 'Normal' legend type. Following is an example of changing legend type form 'Bar_Vertical' to 'Normal'.



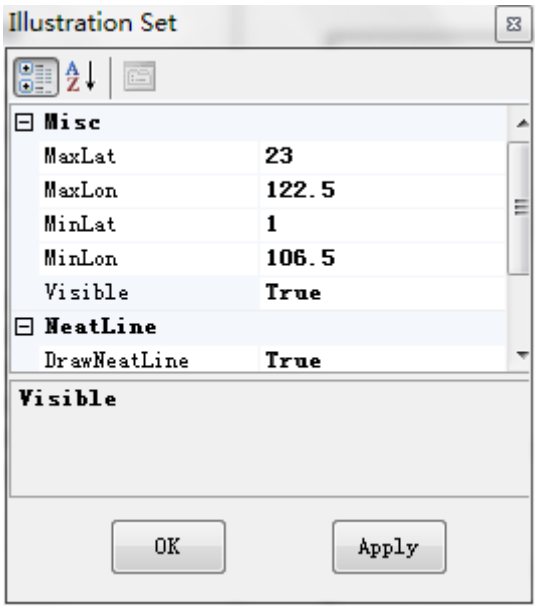
Mask Layer

Mask layer could be set to avoid drawing graphic outside the mask extent. Set ‘SetMaskLayer’ to ‘True’, and then select a mask layer from ‘MaskLayer’ list. But firstly you have to have at least one mask layer in ‘Layers’ list. Only polygon layer could be used as mask layer. ‘china.shp’ file in ‘Map’ folder under the software installation path can be used. Open it, and then select it in ‘MaskLayer’ list. Don’t use very complex polygon layer as mask layer, it will slow down the software obviously.

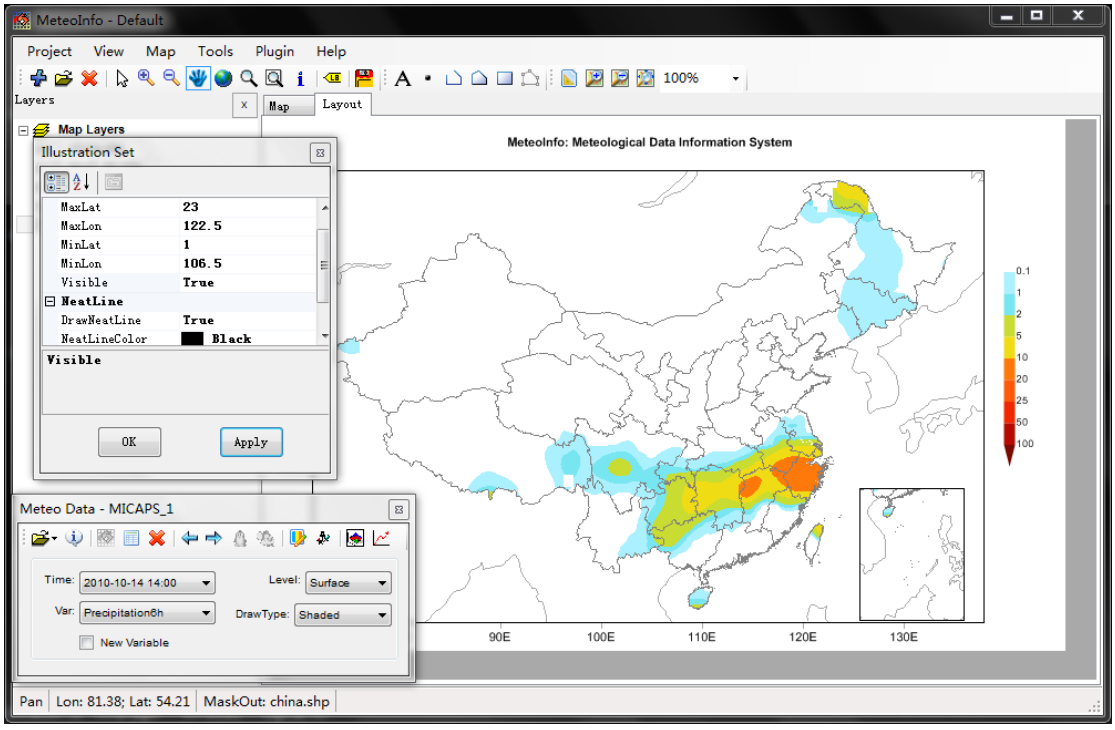


Illustration

Click ‘Illustration’ sub-menu item under ‘Map’ menu item to open ‘Illustration Set’ dialog.



Select ‘Visible” to ‘True’, a default illustration will be plotted. The default setting is the extent of China South Sea area.

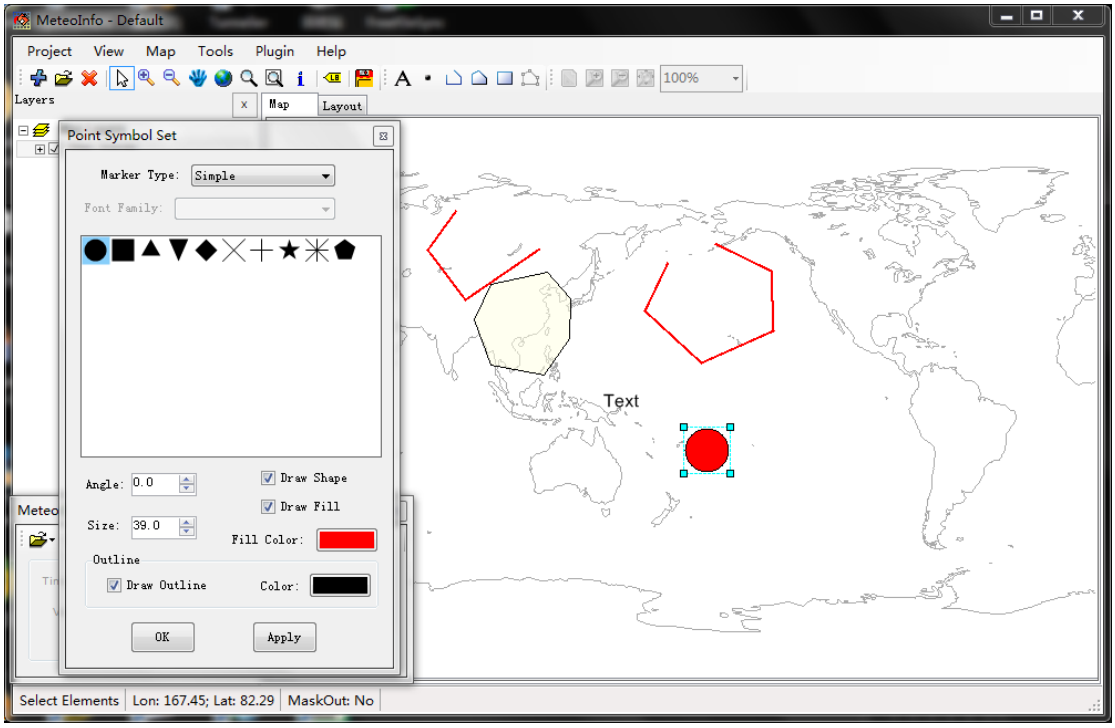


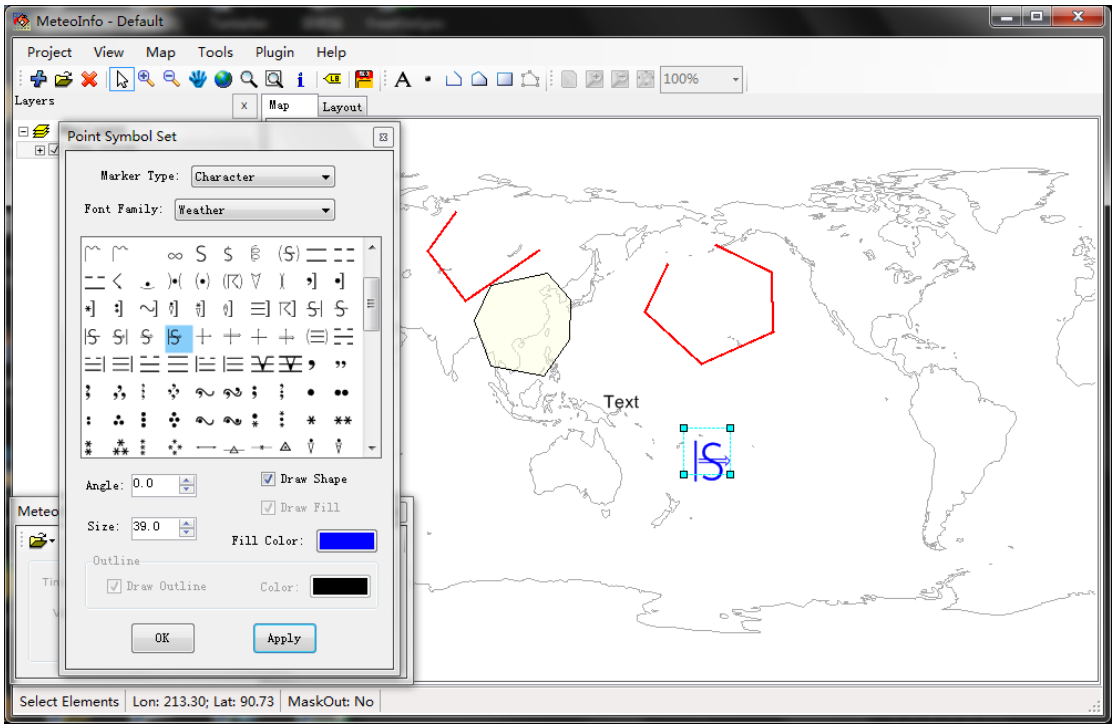
User plotting tools

You can draw label, point, polyline, polygon and rectangle both in ‘Map’ and ‘Layout’ plotting regions. The user plotting objects have same coordinates with the map, and the objects will be moved or re-projected synchronously with the map. The objects plotted on the layout have fixed coordinates which not changed with map view extent.

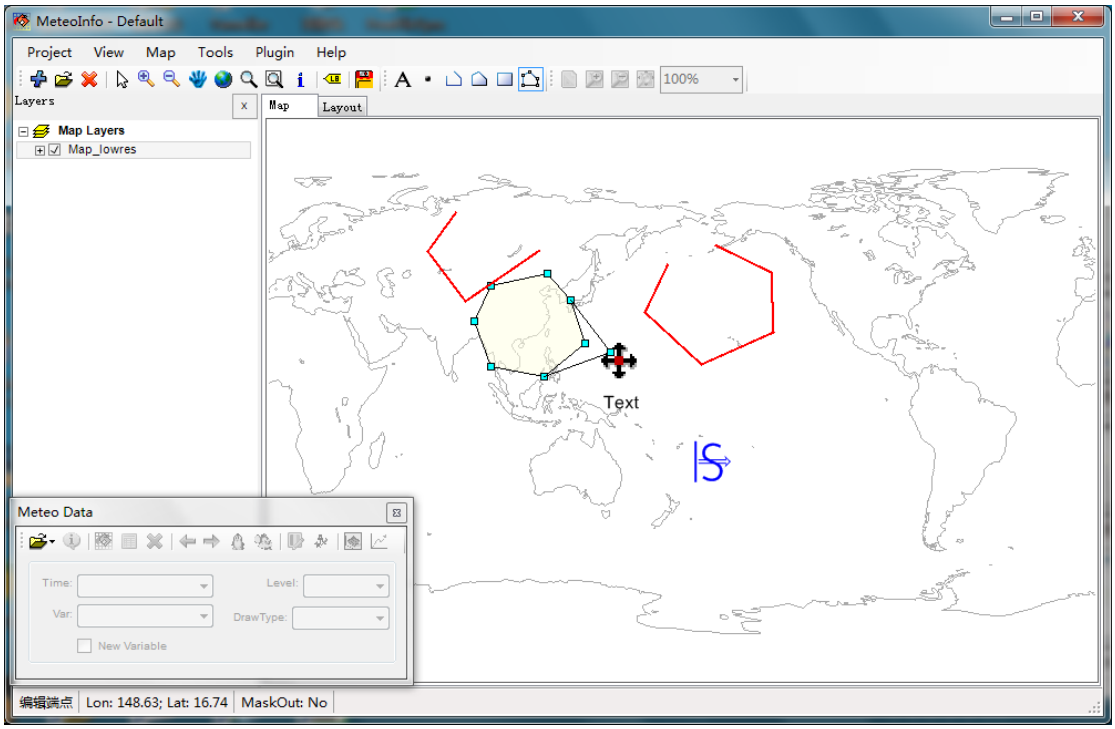
Press one of the tool buttons on the user plotting tool strip, then you can plot label or

geometry objects on map or layout. Label or point object can be plotted by clicking left mouse button in the map or layout plotting regions with the label or point tool selected. For drawing polyline or polygon object, the end points can be added by clicking left mouse button, and the object will be finished by double clicking left mouse button. The objects can be selected by clicking left mouse button using 'Select Elements' tool , then the objects can be moved and resized, or be removed by pressing 'Delete' button on the keyboard. The properties of the object can be edited through the property dialog by double clicking the object.

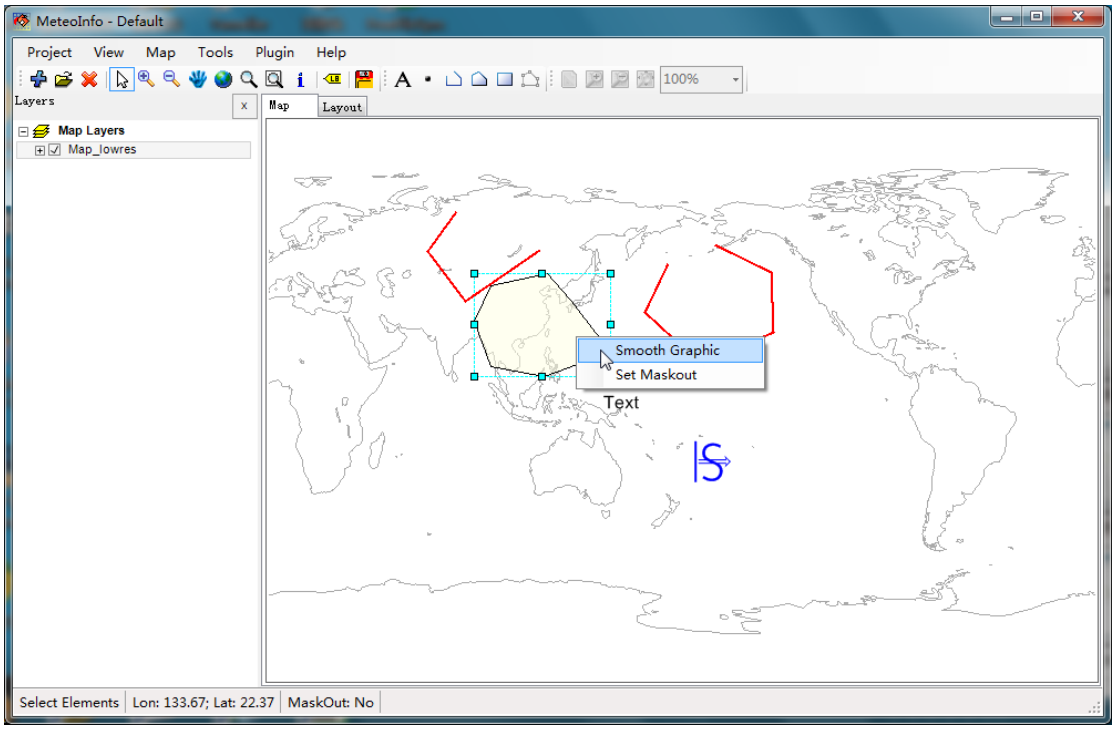


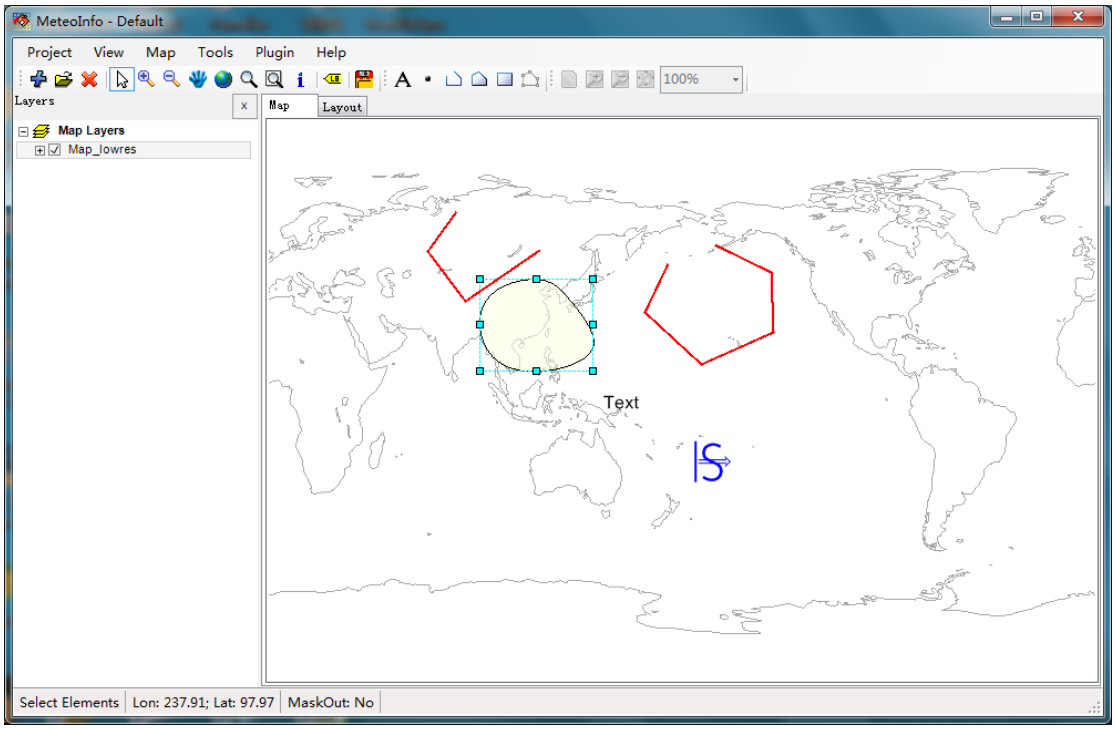


The end points of the polyline or polygon can be moved using ‘Edit Vertices’ tool  when the object is selected.



Polyline or polygon object can be smoothed using the ‘Smooth Graphic’ context menu viewed by clicking right mouse button.



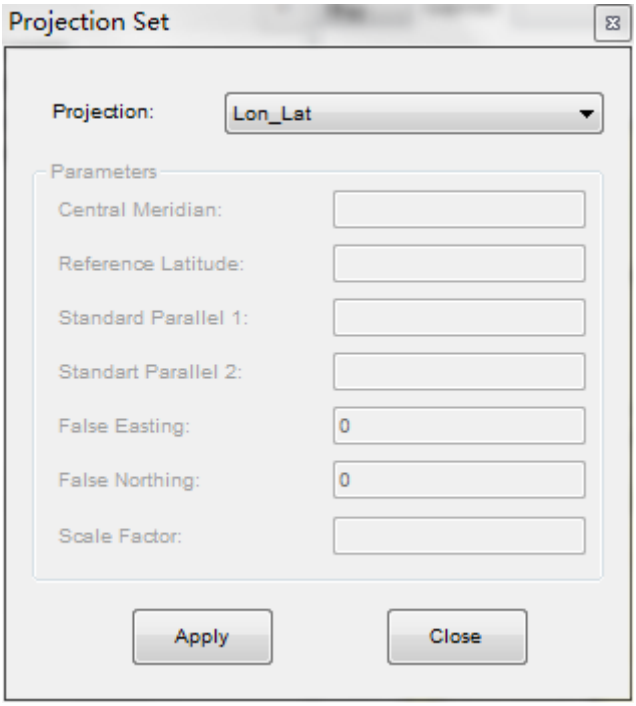


Map Projection

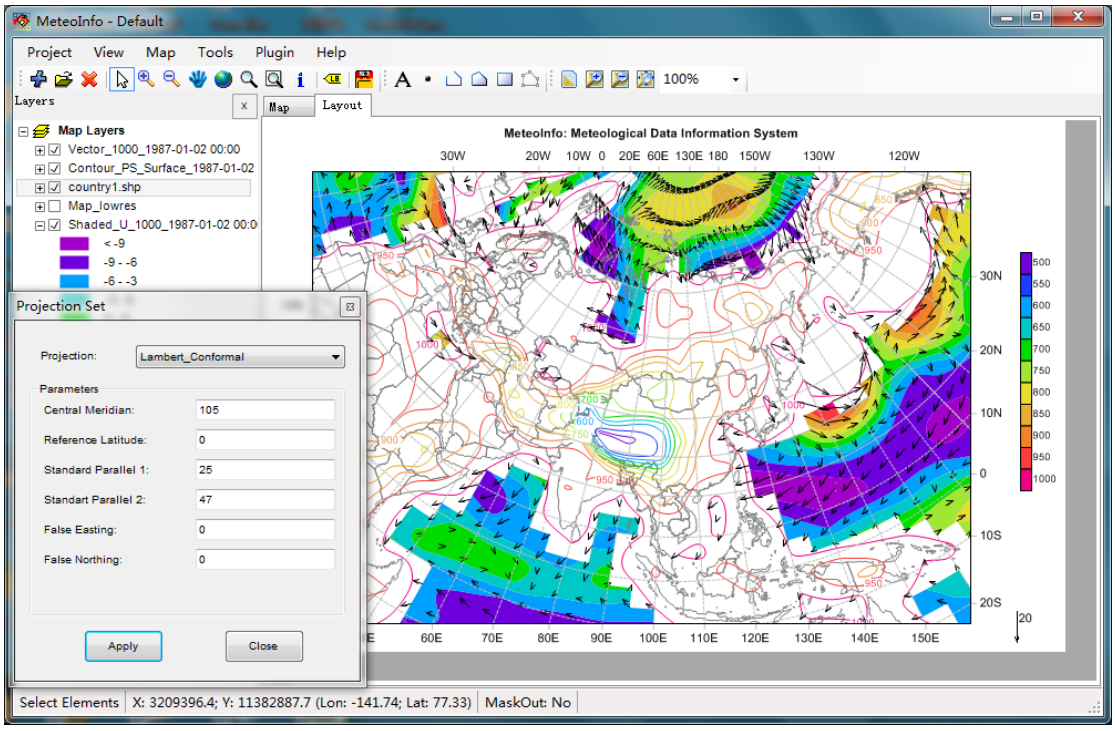
Supported on-the-fly projections:

- Lon_Lat
- Lambert_Conformal
- Albers_Conic_Equal_Area
- North_Polar_Stereographic
- South_Polar_Stereographic,
- Mercator
- Robinson
- Mollweide
- Orthographic
- GeoStationary
- Oblique_Stereographic
- Transverse_Mercator

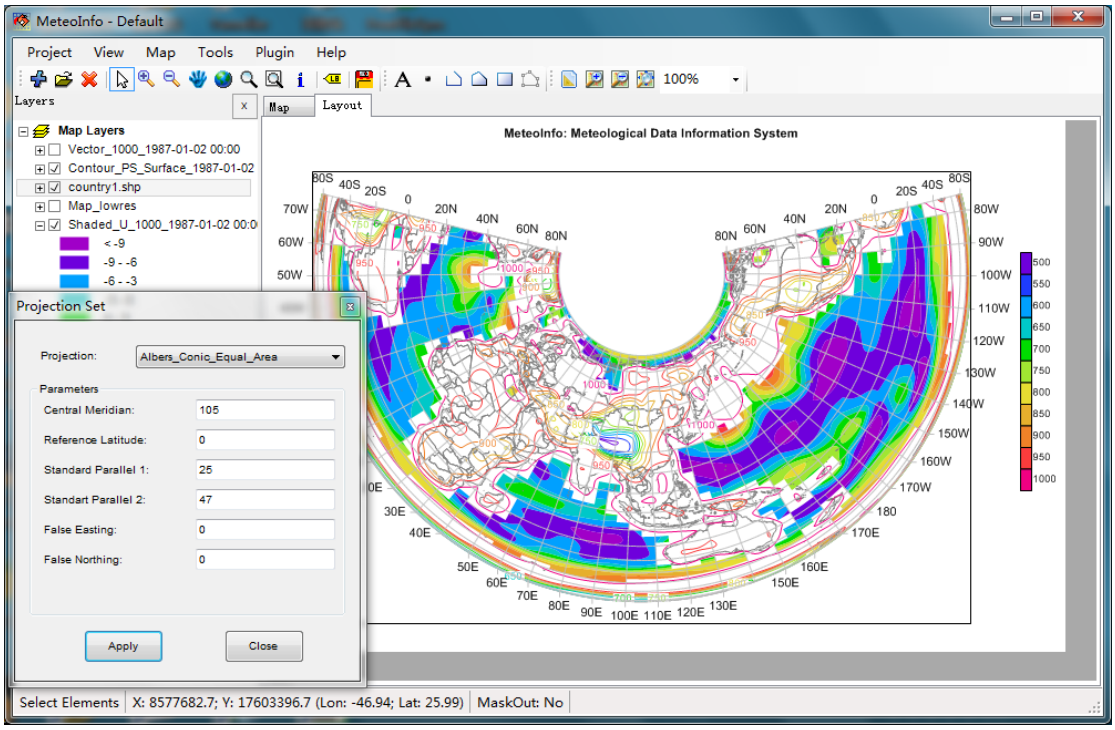
Click ‘Projection’ sub-menu item under ‘Map’ menu item to open ‘Projection Set’ dialog.



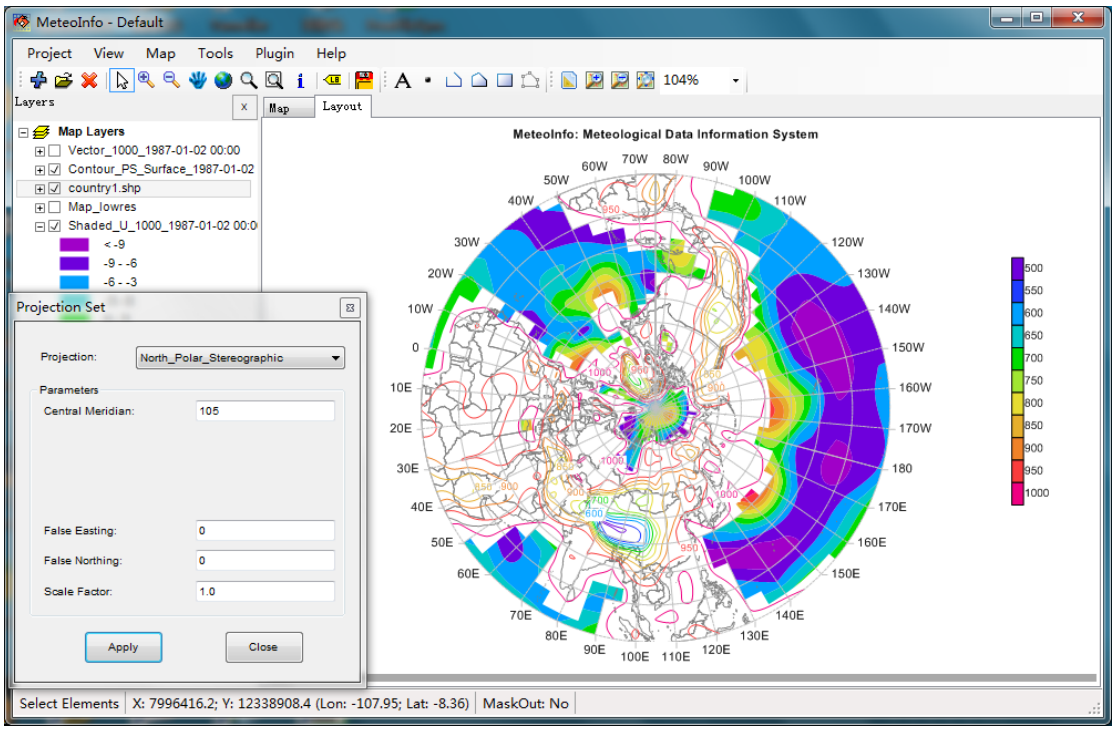
Lambert Conformal projection.



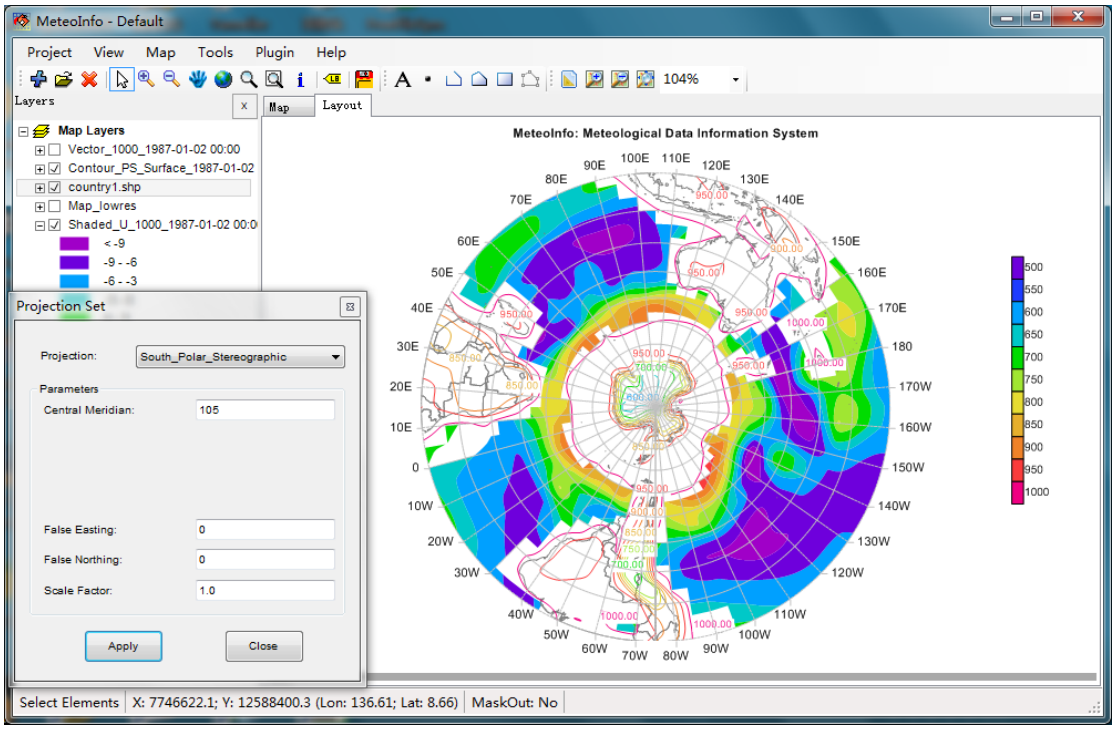
Albers Conic Equal Area Projection.



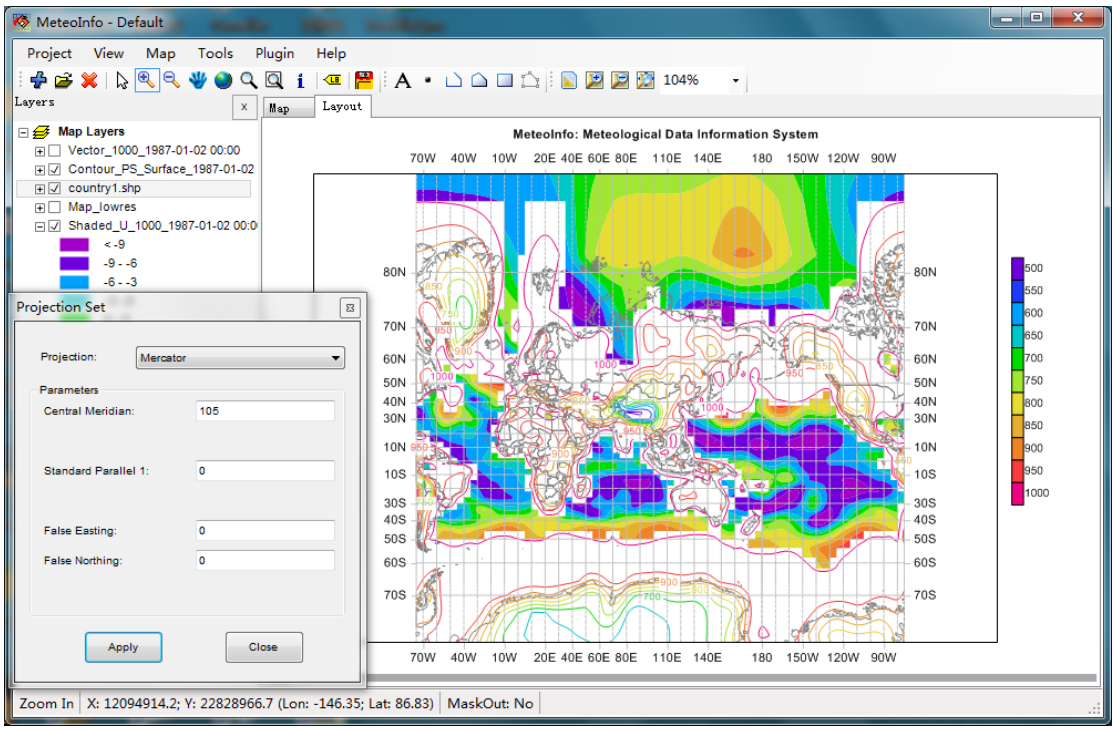
North_Polar_Stereographic



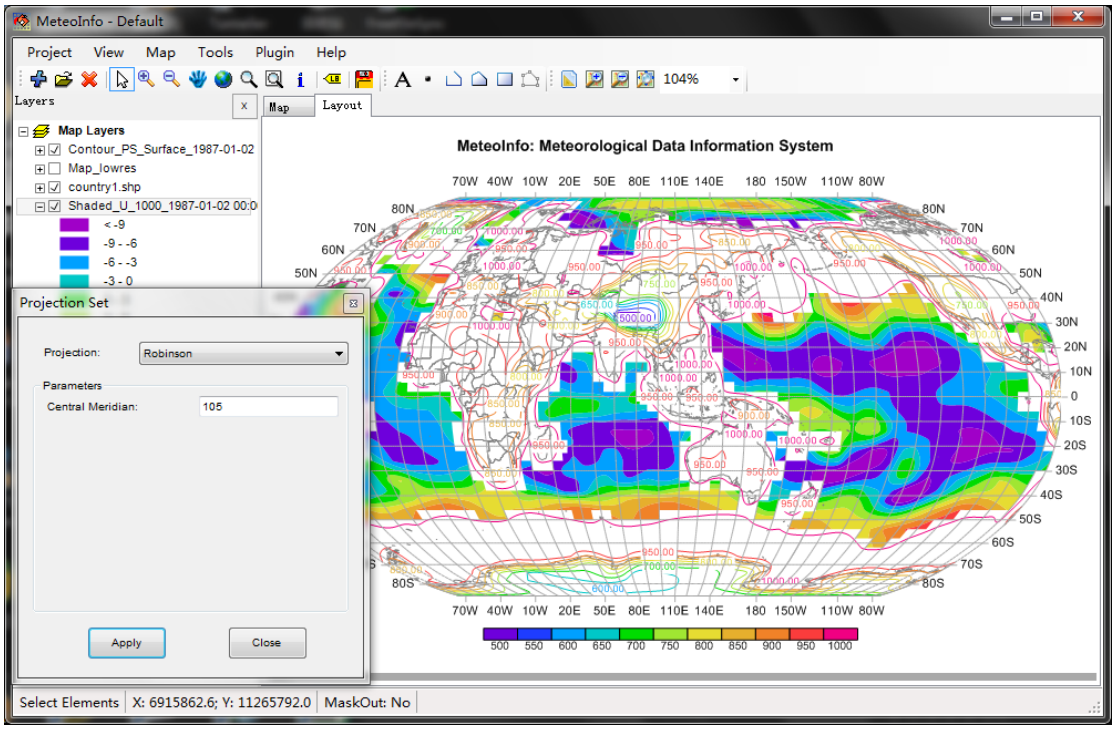
South_Polar_Stereographic



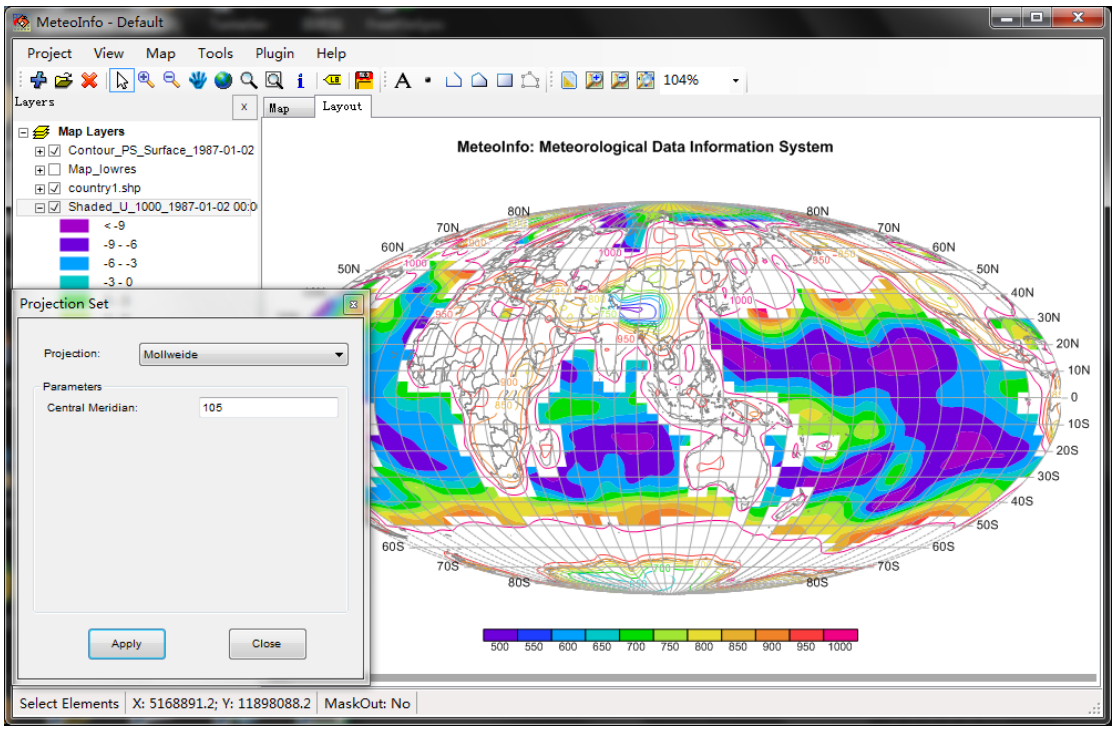
Mercator



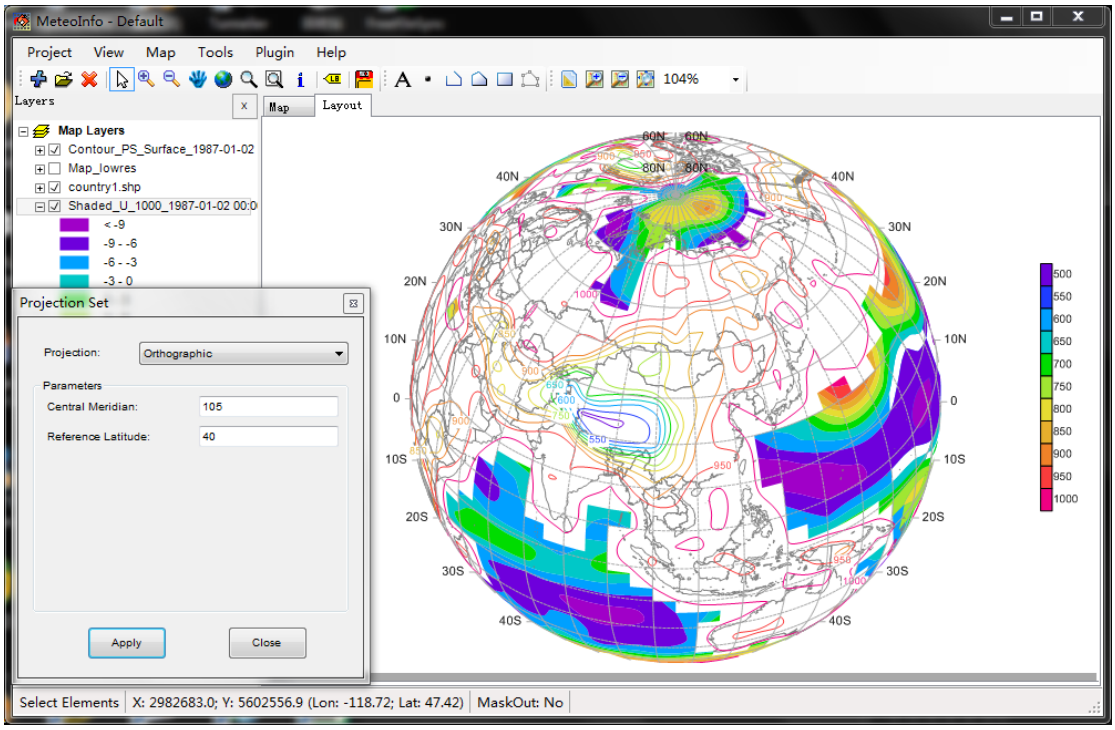
Robinson



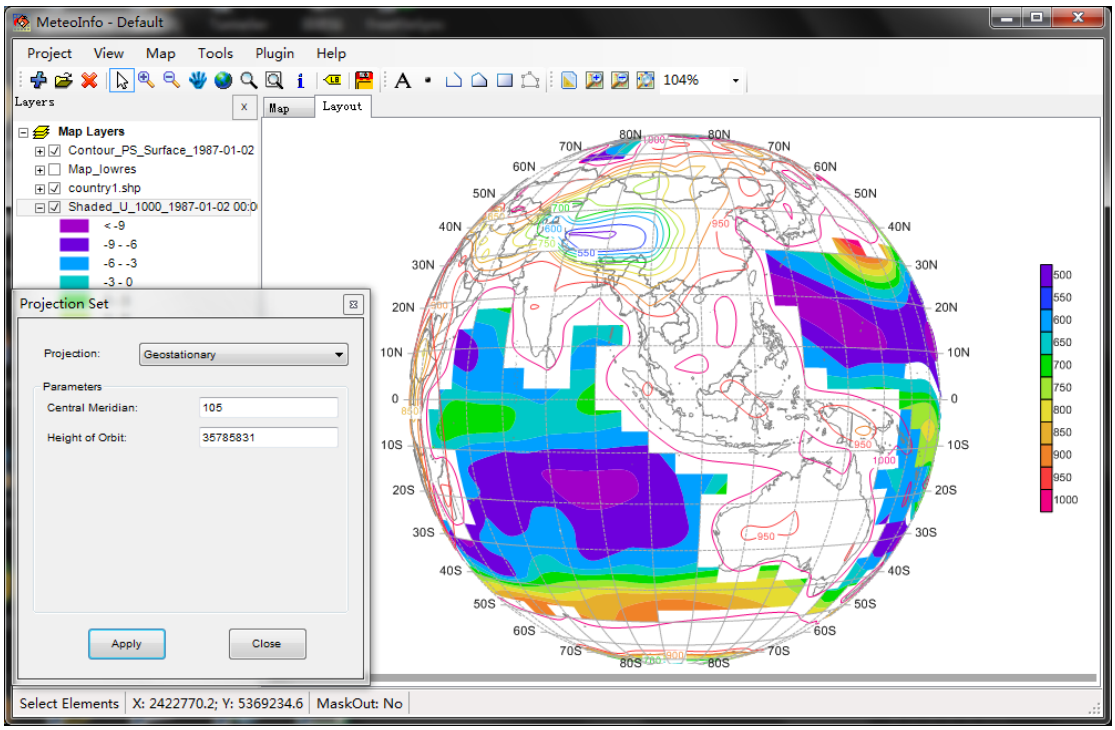
Mollweide



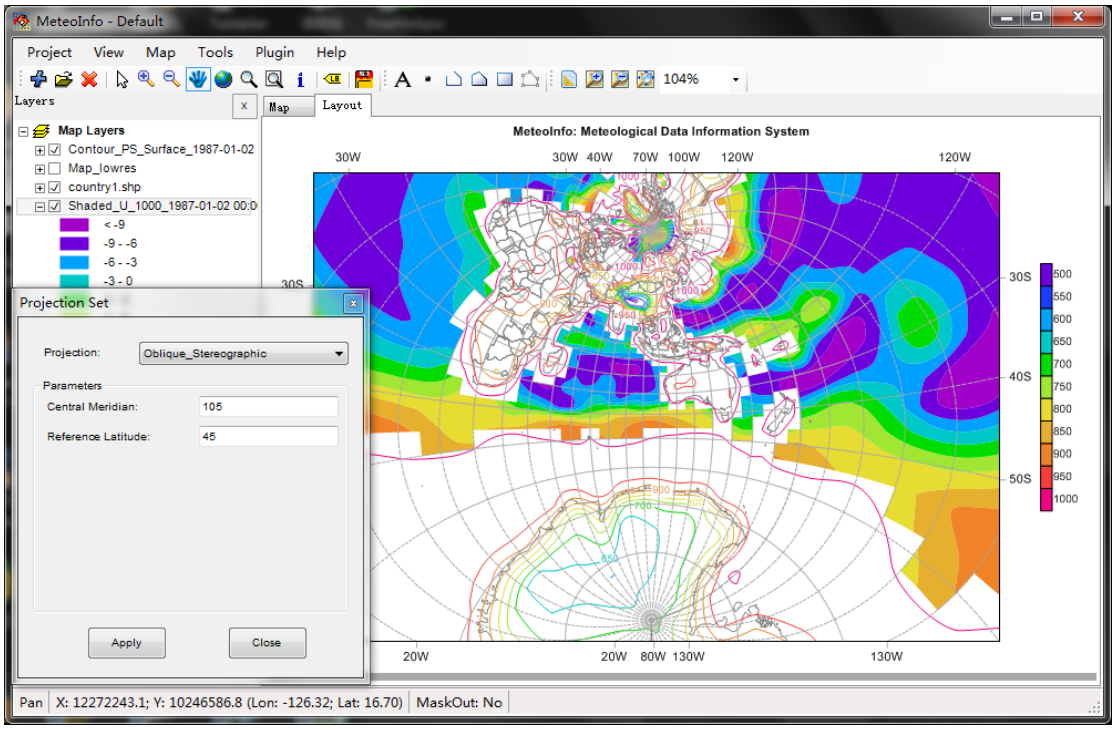
Orthographic



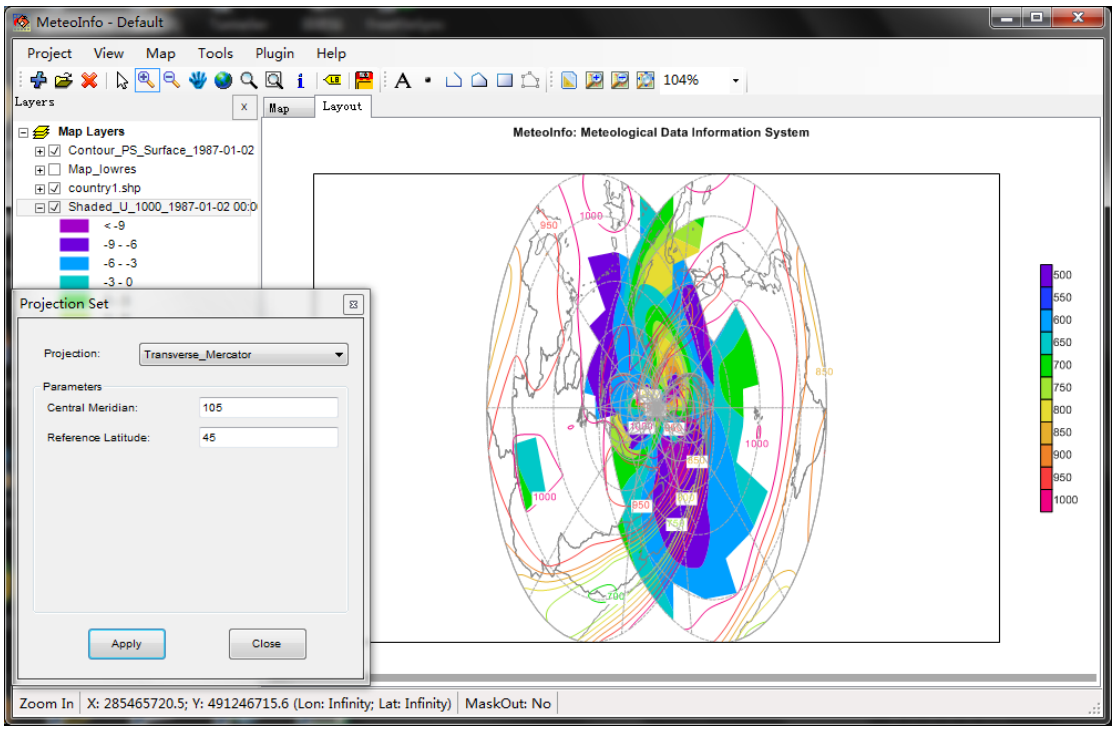
GeoStationary



Oblique_Stereographic



Transverse_Mercator

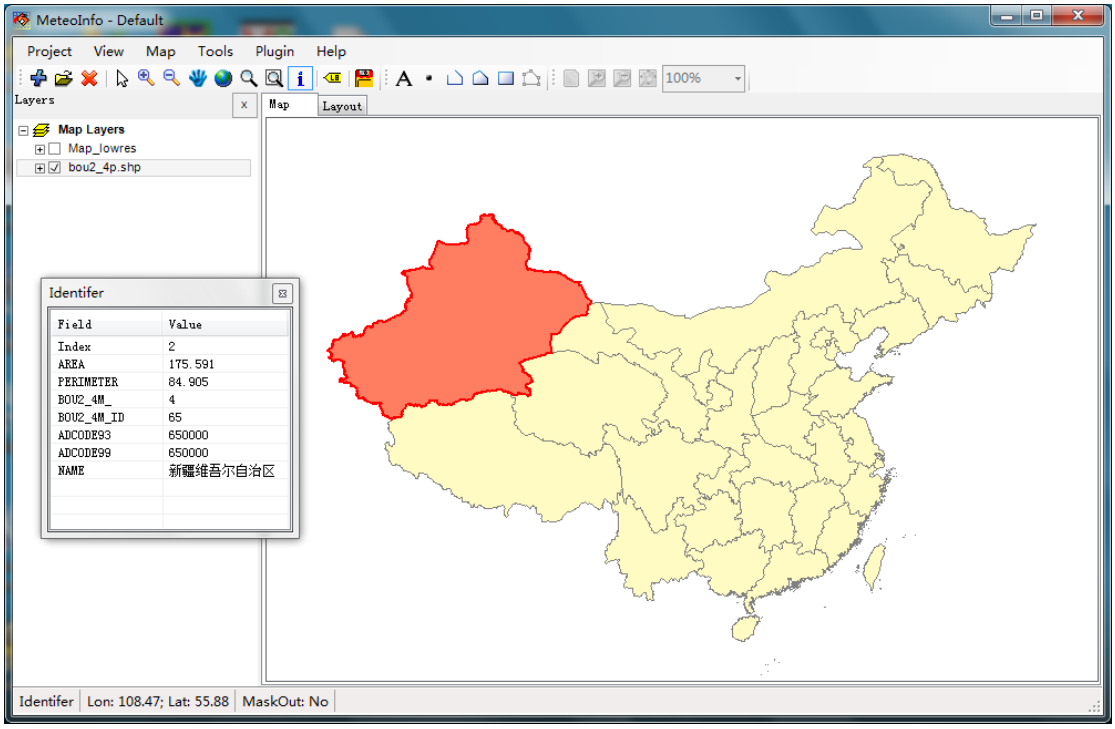


Tools

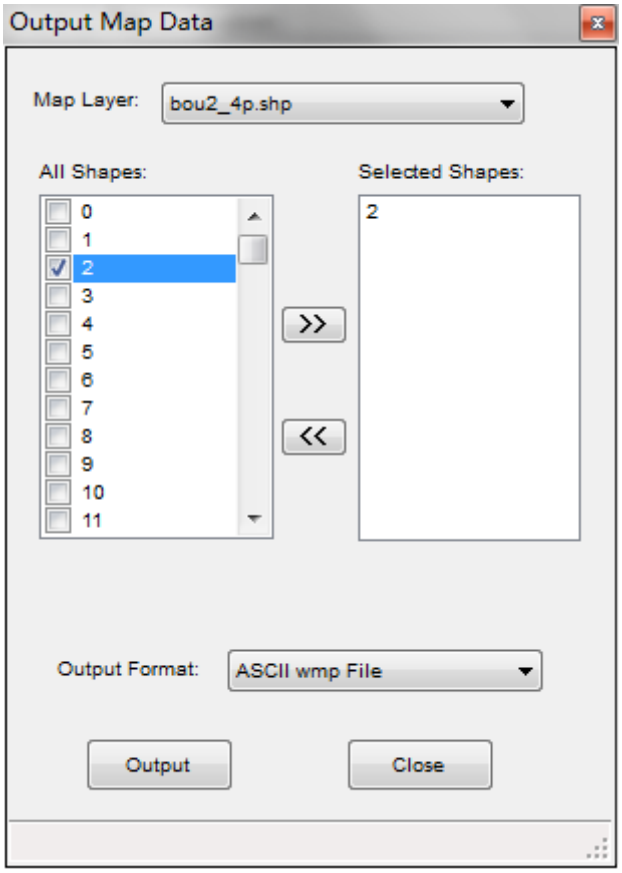
Output Map Data

This function is designed to convert selected shapes in a map layer to other format files which contains border data of the shapes. Following example is to create an ASCII wmf file with latitude/longitude border data of Xinjiang province of China.

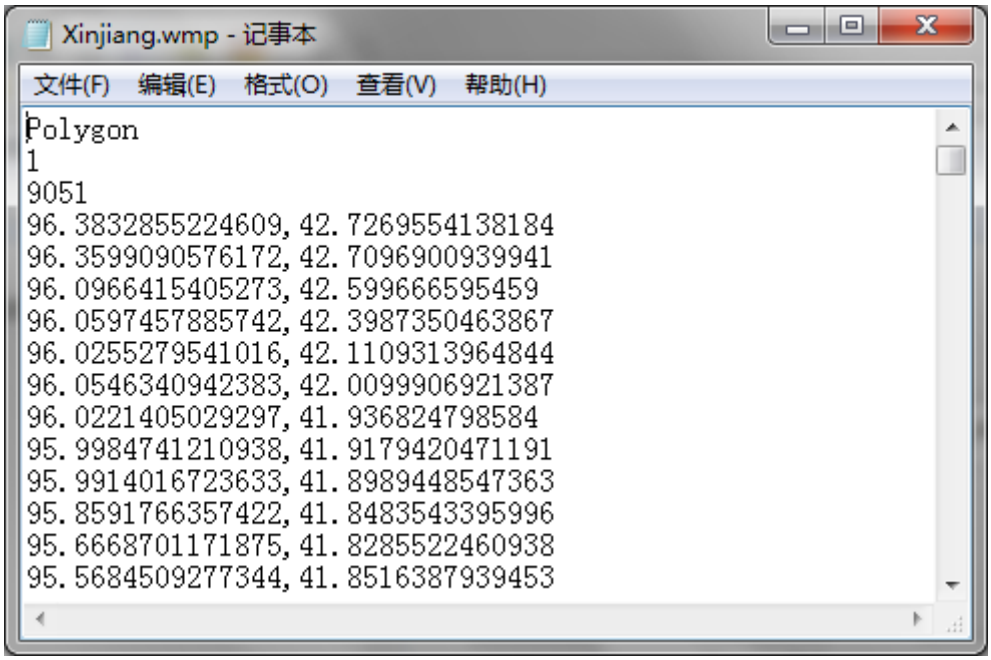
Add and select ‘bou2_4p.shp’ layer, and check the index of the Xinjiang shape with ‘Identifer’ tool button. The index is 2 in this case.



Click ‘Output Map Data’ sub-menu item under ‘Tools’ menu item to open ‘Output Map Data’ dialog.

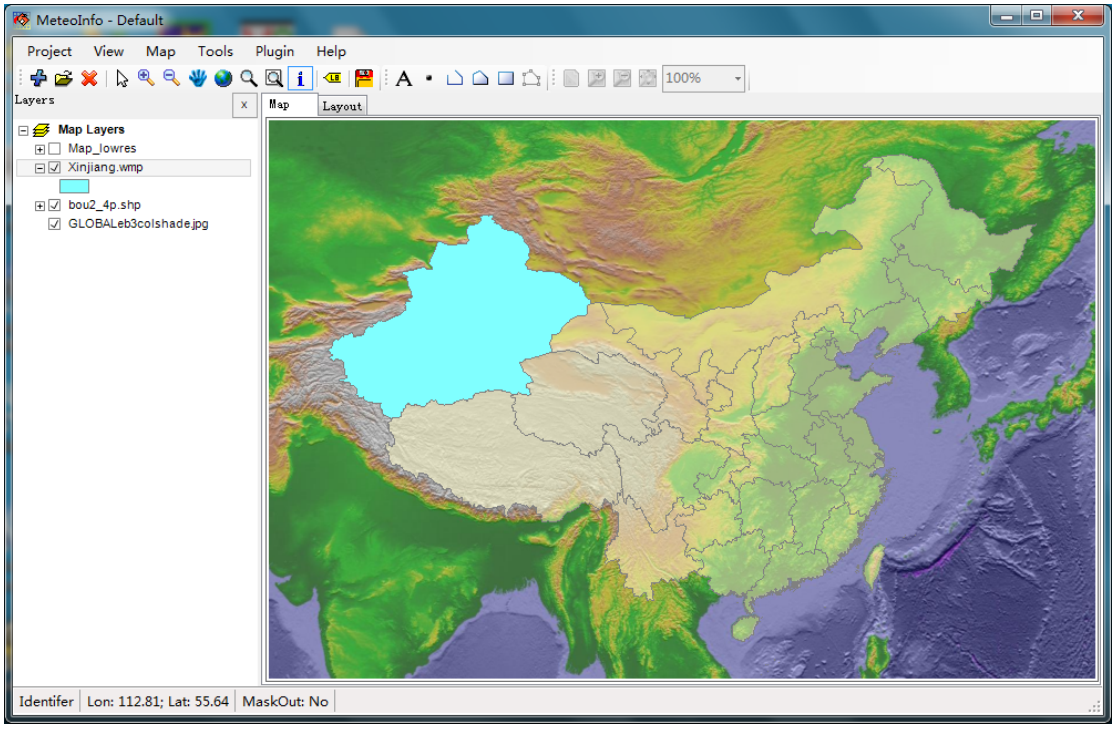


Select ‘Map Layer’ as ‘bou2_4p.shp’ and select shape index 2 (Xinjiang province). Set ‘Output Format’ as ‘ASCII wmp File’ and then press ‘Output’ button. Set output file name ‘Xinjiang.wmp’ in ‘Save As’ dialog then the border longitude/latitude data will be saved in the file.

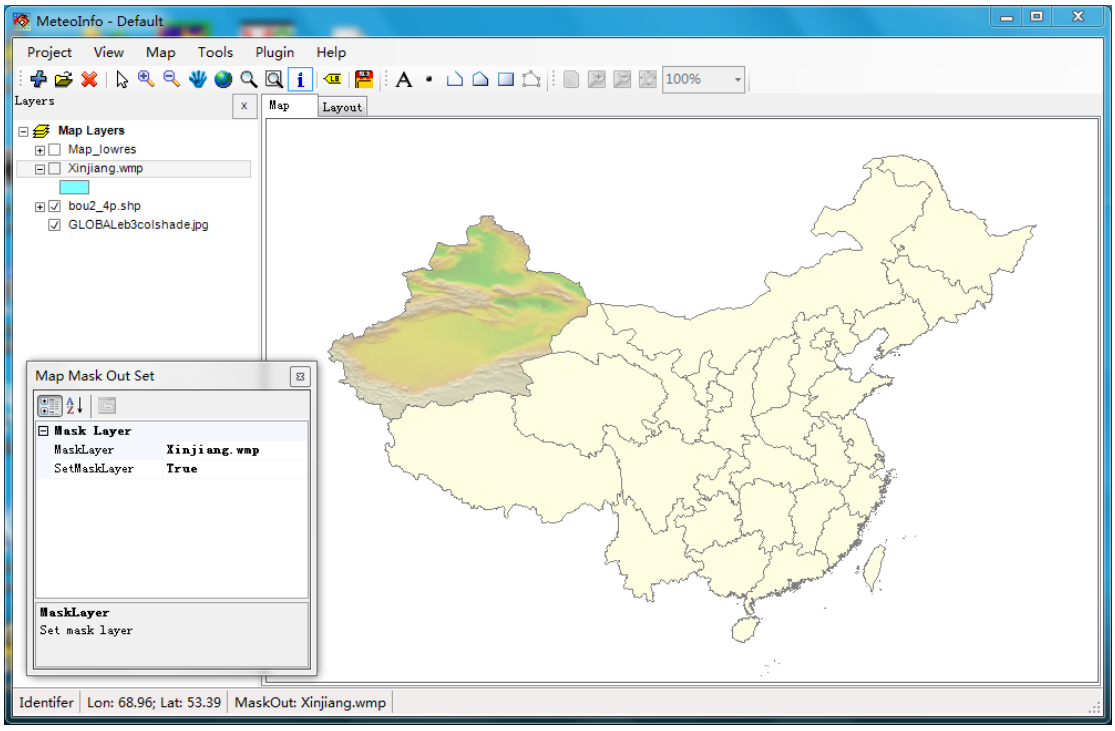


In the ASCII wmp file, first line is the shape type (Point, Polyline or Polygon). Second line is total shape number. The data of each shape were written with point number and longitude/latitude pairs. Longitude and latitude data were separated with comma.

wmp file could be opened in MeteoInfo as a layer.



Because 'Xinjiang.wmp' is a polygon layer, it can be the mask layer.



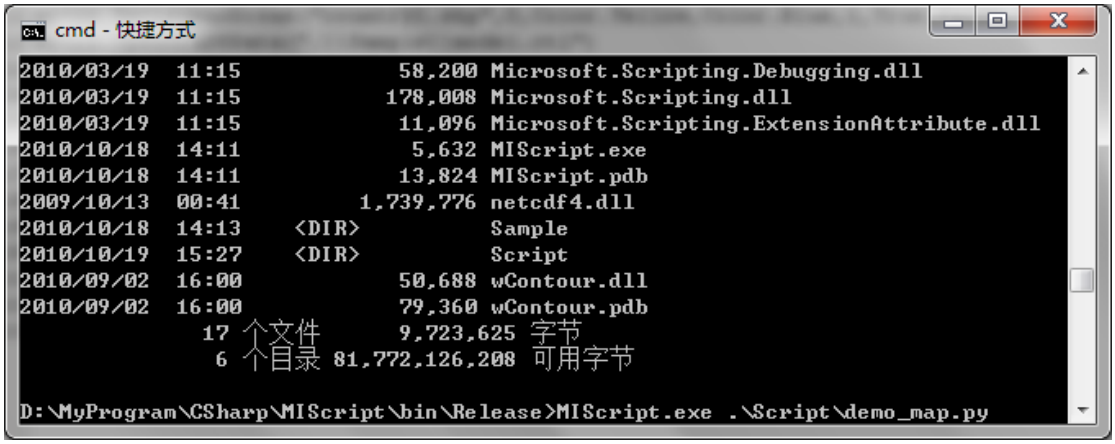
MeteoInfo Script

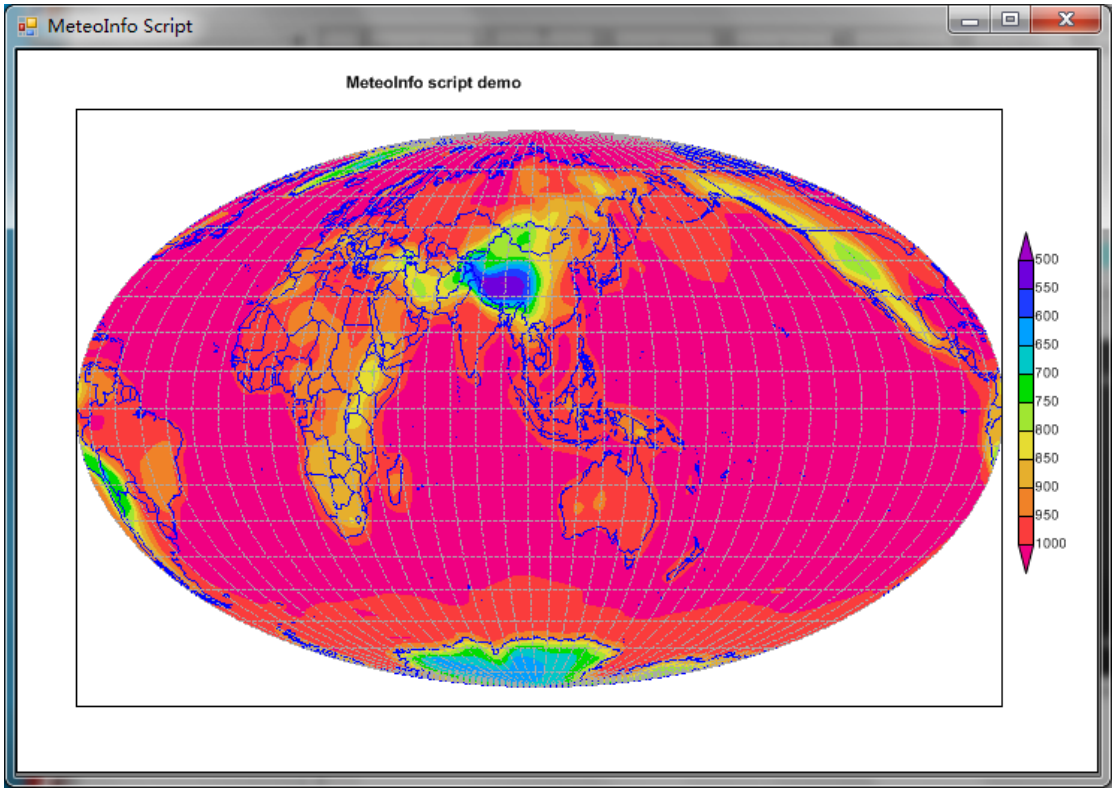
MeteoInfo could be run and output maps automatically using script application

'MIScript.exe' which could be found in MeteoInfo installation folder. The script language is IronPython (http://ironpython.codeplex.com/).

There are some demo scripts in the 'Script' folder of the software.

The script should be run in command window as following.





The script content of the above example is following.

```
import clr
clr.AddReferenceByPartialName("System.Windows.Forms")
clr.AddReferenceByPartialName("System.Drawing")
from System.Windows.Forms import *
from System.Drawing import *
clr.AddReference("MeteoInfoC.dll")
```

```
from MeteoInfoC import *

myApp = MIApp()
myApp.OpenLayer(".\\Map\\country1.shp")
myApp.SetLegendBreak("country1.shp",0,Color.Yellow,Color.Blue,1,True,False,True)
myApp.OpenGrADSData(".\\Sample\\model.ctl")
myApp.SetDrawType("shaded")
t = 2
myApp.TimeIndex = t
myApp.Display("PS")
myApp.MoveLayerToTop("country1.shp")
myApp.SetTitle("MeteoInfo script demo")
myApp.ProjectLayers("+proj=moll+lon_0=105")
myApp.MapLayout.Refresh()
#myApp.SaveFigure(".\\try.png")
Application.Run(myApp)
```

Using CLR and .NET Framework Classes

IronPython brings together the elegant Python programming language and the powerful .NET platform. Common Language Runtime (CLR) forms the core execution engine of .Net platform. Along with the runtime comes a vast array of libraries and classes, collectively known as the framework classes. IronPython is a full .NET language and almost all features of the .NET platform can be used.

MeteoInfo script is using IronPython engine, so you need some basic knowledge of Python and .Net framework for programming MeteoInfo script. “IronPython in Action” is a good reference book on it.

To use the classes of .NET, import CLR first and then add references to the .NET assembly. For example, the classes included in System, System.Windows.Forms and System.Drawing name spaces will be available using following codes.

```
import clr
clr.AddReferenceByPartialName("System.Windows.Forms")
clr.AddReferenceByPartialName("System.Drawing")
from System.Windows.Forms import *
from System.Drawing import *
```

Using Functions of MeteoInfo

The main functions of MeteoInfo are packed in a .NET class library ‘MeteoInfoC.dll’, which includes a ‘MIApp’ class for writing MeteoInfo script simply.

Add ‘MeteoInfoC.dll’ as a reference.

```
clr.AddReference("MeteoInfoC.dll")
```

```
from MeteoInfoC import *
```

Create an object of 'MIApp' class.

```
myApp = MIApp()
```

Then the functions of 'MIApp' class could be used in the script.

Open Data

Several open data functions are provided to open multi-source spatial or meteorological data.

- Open map or image data file: OpenLayer(string aFile)
- Open GrADS data file: OpenGrADSData(string aFile)
- Open NetCDF data file: OpenNCData(string aFile)
- Open GRIB data file: OpenGRIBData(string aFile)
- Open MICAPS data file: OpenMICAPSData(string aFile)
- Open ARL packed meteorological data file: OpenARLData(string aFile)
- Open Lon/Lat station data file: OpenLonLatData(string aFile)
- Open HYSPLIT concentration data file: OpenHYSPLITConc(string aFile)

Example:

```
myApp.OpenGrADSData(".\\Sample\\model.ctl")
```

Set Dimensions

Only horizontal data could be operated by MIApp class at present. The ability to operate other data such as section or one dimension data will be added later. Time index and Level index could be set at present.

- Set time index: myApp.TimeIndex = i
- Set level index: myApp.LevelIndex = i

The default dimensions are 0. The grid or station data will be read according to the dimension setting.

Set Drawing Type

The drawing type will be set using SetDrawType(string drawTypeString) function. The 'drawTypeString' parameter could be assigned as following:

- "contour"
- "shaded"
- "grid_fill"
- "grid_point"
- "vector"
- "barb"

- “streamline”
- “station_point”
- “weather_symbol”
- “station_model”
- “station_info”
- “traj_line”
- “traj_point”
- “traj_startpoint”
- “image”
- “raster”

Example:

```
myApp. SetDrawType ("shaded")
```

Read the Meteorological Data

There are two important data classes in MeteoInfo: GridData and StationData. They are the main dataset for analysis and visualization.

- Get grid data set: GetGridData(string varName) , return a GridData object.
- Get station data set: GetStationData(string varName) , return a StationData object

The parameter of the above functions is variable name of the meteorological data.

Example:

```
UGridData = myApp.GetGridData("U")
rainSTData = myApp.GetStationData("Precipitation6h")
```

Plot the Meteorological Data

Some override ‘Display’ functions are used to plot the grid or station data set.

- Display(string varName)
- Display(GridData aGridData)
- Display(string U, string V)
- Display(string U, string V, string varName)
- Display(GridData UGridData, GridData VGridData)
- Display(GridData UGridData, GridData VGridData, GridData XGridData)
- DisplayWind(string windDir, string windSpeed)
- DisplayTraj()

Example:

```
myApp.Display("PS")

UGridData = myApp.GetGridData("U")
VGridData = myApp.GetGridData("V")
```

```
myApp.Display(UGridData, VGridData)
```

Calculate the Meteorological Data

The calculation operators of '+', '-', '*' and '/' are supported for grid or station data analysis. Also some math functions in 'DataMath' class could be used to do data calculation. The current functions are listed bellow. The 'MeteoInfoC.Data' name space should be added which includes 'DataMath' class.

- Abs
- Acos
- Asin
- Atan
- Cos
- Sin
- Tan
- Exp
- Pow
- Sqrt
- Log
- Log10
- Magnitude
- Cdiff
- Hcurl
- Hdivg

Example:

```
...
#Import the classes from 'MeteoInfoC.Data' name space
from MeteoInfoC.Data import *
...
#Get U/V components grid data
UGridData = myApp.GetGridData("U")
VGridData = myApp.GetGridData("V")
#Calculate wind speed from U/V components
Speed = DataMath.Magnitude(U, V)
#Or
Speed = DataMath.Sqrt(U * U + V * V)
#Plot wind speed as shaded layer
myApp.SetDrawType("shaded")
myApp.Display(Speed)
```

Other Functions

- ProjectLayers(string projStr)

  Project the plot by Proj4 string parameter. Please check Proj4 help document to know how to set projection string. Also some websites have the projection information, such as http://remotesensing.org/geotiff/proj_list/ .

  Example:

  myApp.ProjectLayers("+proj=moll+lon_0=105")

- Zoom(double minX, double maxX, double minY, double maxY)

  Zoom the plot to the extent. Width and Height ratio of the map view will not be changed.

  Example:

  myApp.Zoom(75, 135, 15, 55)

- ZoomEx(double minX, double maxX, double minY, double maxY)

  Zoom the plot to the exact extent. Width and Heigh ratio of the map view will be changed to fit the extent.

  Example:

  myApp.ZoomEx(0, 360, -90, 90)

- SetMaskout(string layerName)

  Set maskout layer by layer name.

  Example:

  myApp.SetMaskout("china.shp")

- SetIllustration(bool visible)

  Set illustration visible or not.

  Example:

  myApp.SetIllustration(True)

- SaveFigure(string aFile)

  Save the plot to an image file.

  Example:

  myApp.SaveFigure(".\\try.png")

- SetInterpolation(double minX, double maxX, double minY, double maxY, int xNum, int yNum, string aInterMethod, float radius, int minNum)

  Set interpolation parameters for interpolating station data to grid data.

  Example:

  myApp.SetInterpolation(60,140,-20,60,160,160,"IDW_Radius",1,1)

- SetLegendScheme(string legendFile)

  Set legend scheme by the legend file.

  Example:

  myApp.SetLegendScheme(".\\Legend\\rain.lgs")

- SetLegendBreak(string layerName, int brkIdx, Color aColor, Color outlineColor, float outlineSize, bool drawOutline, bool drawFill, bool drawShape)

  Set a legend break of the legend scheme of the layer by layer name.

  Example:

  myApp.SetLegendBreak("country1.shp",0,Color.Yellow,Color.Gray,1,True,False,True)

- SetTitle(string aTitle)

  Set the plot's title.

  Example:

  myApp.SetTitle("MeteoInfo script demo - Grid data calculation")
- MoveLayerToTop(string layerName)

  Move the layer to the top of the layers, then the layer will be plotted on the top of other layers.

  Example:

  myApp.MoveLayerToTop("country1.shp")