Unidata IDV Workshop

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Unidata IDV Workshop for version 3.1

This workshop and the associated materials are designed to give you an introduction to installing, using, and configuring the Integrated Data Viewer (IDV). Each section is designed to provide an overview of the topic covered with examples and exercises for you to complete. The conventions used in this document can be found here. Most of the datasets used in these exercises are accessible on remote servers. For accessing the local datasets, see Installing local datasets.
The workshop is broken down into the following major sections:

1 Overview of the Integrated Data Viewer

This section provides an overview of the IDV, its purpose and its capabilities.

1.0 What is the Integrated Data Viewer?

Overview
Unidata's Integrated Data Viewer (IDV) is a Java -based software framework for analyzing and visualizing geoscience data. The IDV release includes a library and a reference application made from that software. It is based on VisAD (http://www.ssec.wisc.edu/~billh/visad.html), an open-source, Java library for building interactive and collaborative visualization and analysis tools, as well as other Java-based utility packages.

The IDV is being developed at the Unidata Program Center (UPC), part of the University Corporation for Atmospheric Research, Boulder, Colorado, under funding provided by the National Science Foundation. The software is freely available under the terms of the GNU Lesser General Public License.

The IDV brings together the ability to display and work with satellite imagery, gridded data (for example, numerical weather prediction model output), surface observations (METARs), upper air soundings, NWS NEXRAD Level II and Level III RADAR data, NOAA National Profiler Network data and GIS data, all within a unified interface. The IDV "reference application" provides many of the standard 2-D data displays that other Unidata packages (e.g. GEMPAK and McIDAS) provide. It also provides 3-D views of the atmosphere and allows users to interactively slice, dice, and probe the data, creating cross-sections, profiles, animations and value read-outs of multi-dimensional data sets. Computations and displays of built-in and user-supplied formula-based derived quantities are supported. A feature unique to the IDV is an integrated HTML display that can be used to create HTML-based users' interfaces to drive the displays. You can also embed fully-interactive 2D and 3D IDV displays in HTML documents, when viewed with the IDV. The IDV User's Guide describes the IDV reference application. The IDV software library can be easily be used and extended to create custom geoscience applications beyond the atmospheric science realm. This customization allows new applications to be tailored to specific datasets and provide customized user interfaces for different tasks. One example of a specialized IDV application is the GEON IDV (http://geon.unavco.org/unavco/IDV_for_GEON.html)

2 Starting the IDV/GUI Basics

In this section, you will learn about starting the IDV from a command line and gain a basic understanding of the IDV Graphical User Interface (GUI).
If you have not yet installed the IDV, see the Installing the IDV section before starting.

2.0 Starting the IDV for the first time

Once the IDV has been installed, you can start it from the command line.
Starting the IDV using the `runIDV` script.

1. Open a terminal window (if you do not have one open already) and change to the directory where the IDV was installed. (e.g., `/home/idv/IDV_3.1`)

   ```
   cd  /home/idv/IDV_3.1
   ```
2. Print out the `runIDV` script.

   On Unix/Mac:

   ```
   cat  runIDV
   ```

   On Windows:

   ```
   type  runIDV.bat
   ```

   You should see a printout that ends like this:

   On Unix:

   ```
   command="java -Xmx${idv_memory}m -Didv.enableStereo=false -jar ${dirname}/idv.jar $*"
   ```

   and then some code to find the version of java to use.

   On Windows:

   ```
   jre injava -Xmx%idv_memory%m -Didv.enableStereo=false -jar idv.jar %*
   ```

   The script sets up the environment for the IDV to run. Details
3. The IDV has many command line arguments which can be passed to the `runIDV` script. To get a listing, run the following command:

   ```
   ./runIDV -help
   ```

   You should see a printout like this:

   ```
   -help  (this message)
   -properties  <property file>
   -installplugin  <plugin jar file to install>
   -plugin  <plugin jar file, directory, url for this run>
   -noplugins  (don't load plugins)
   -cleardefault  (Clear the default bundle)
   -nodefault  (Don't read in the default bundle file)
   -default  <.xidv file>
   -bundle  <bundle file or url>
   ```

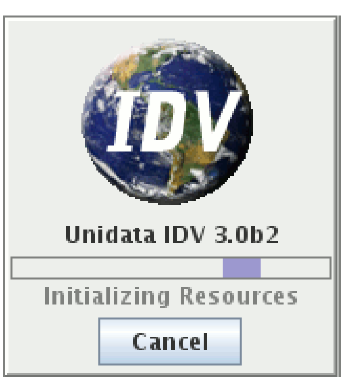
```
-oneinstanceport  <port number>
         (Check if another version of the IDV is running. If so pass command line
          arguments to it and shutdown)
-nooneinstance  (Don't do the one instance port)
-nopref  (Don't read in the user preferences)
-userpath  <user directory to use>
-sitepath  <url path to find site resources>
-nogui  (Don't show the main window gui)
-data  <data source> (Load the data source)
-display  <parameter> <display>
-islinteractive  <scriptfile> (run the isl file in interactive mode);
-islfile <scriptfile>  (Run the IDV Jython script in batch mode)
-image  <image file name> (create a jpeg image and then exit)
-movie  <movie file name> (create a quicktime movie and then exit)
-imageserver  <port number or .properties file>
           (run the IDV in image generation server mode. Support http requests on
-Dpropertyname=value  (Define the property value)
-catalog  <url to a chooser catalog>
-connect  <collaboration hostname to connect to>
-server  (Should the IDV run in collaboration server mode)
-port  <Port number collaboration server should listen on>
-chooser  (show the data chooser on start up)
-printjnlp  (Print out any embedded bundles from jnlp files)
-setfiles  <datasource pattern> <semi-colon delimited list of files>
             (Use the list of files for the bundled datasource)
-currenttime  <dttm> (Override current time for ISL processing)
-listresources  (list out the resource types)
-debug  (Turn on debug print)
-debugmessages  (Turn on language pack debug)
-recordmessages  <Language pack file to write missing entries to>
-trace  (Print out trace messages)
-traceonly  <trace pattern> (Print out trace messages that match the pattern)
```

4. For this workshop, we are going to only use the `-installplugin` argument to load some workshop specific configuration files from a plugin. (You will learn more about plugins later in the workshop).

   To start the IDV for the workshop the first time, type (Cut and Paste tip) :

   ```
   ./runIDV  -installplugin http://www.unidata.ucar.edu/software/idv/plugins/
   ```

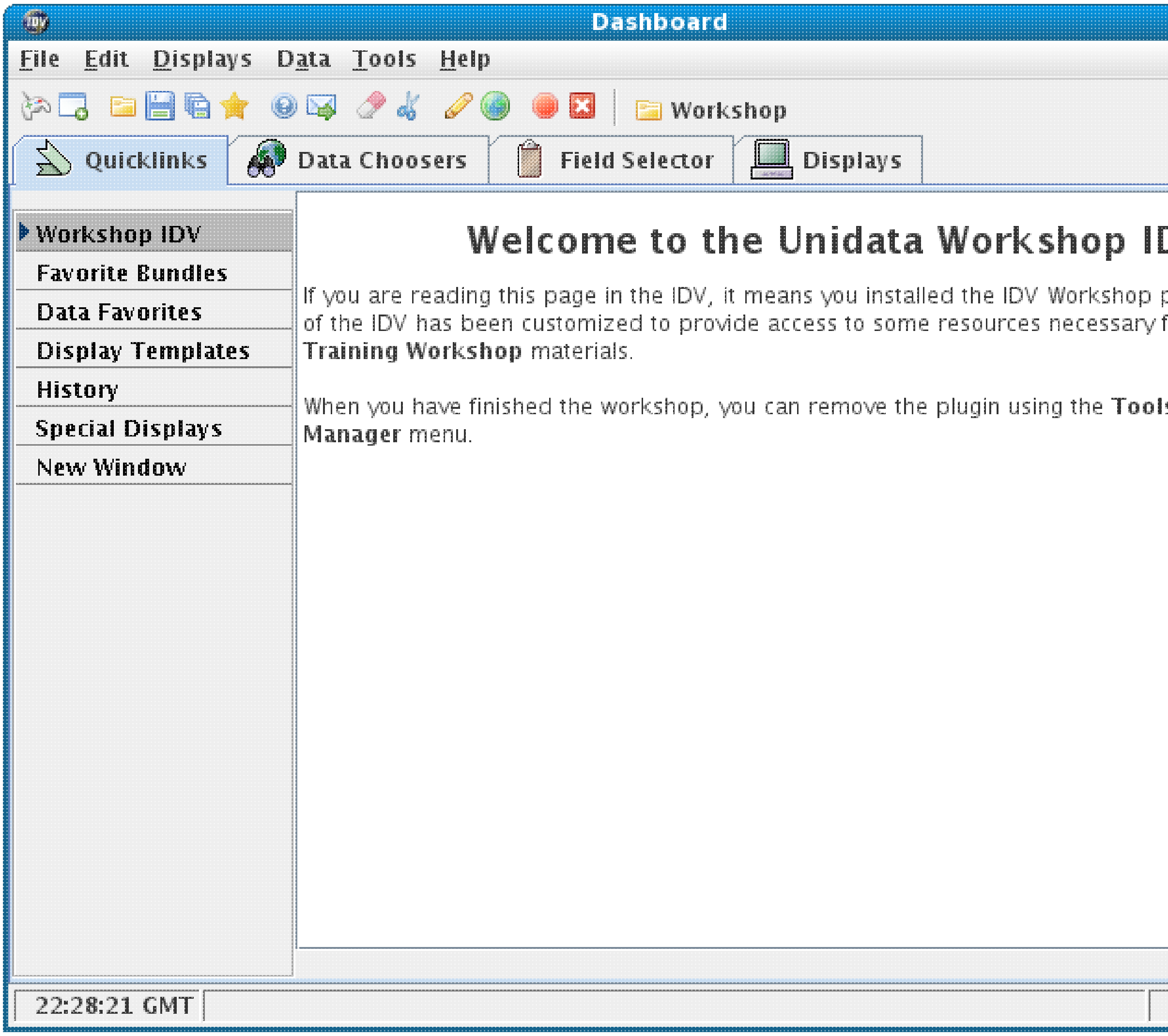
   While the IDV is starting, you will see the following splash screen:

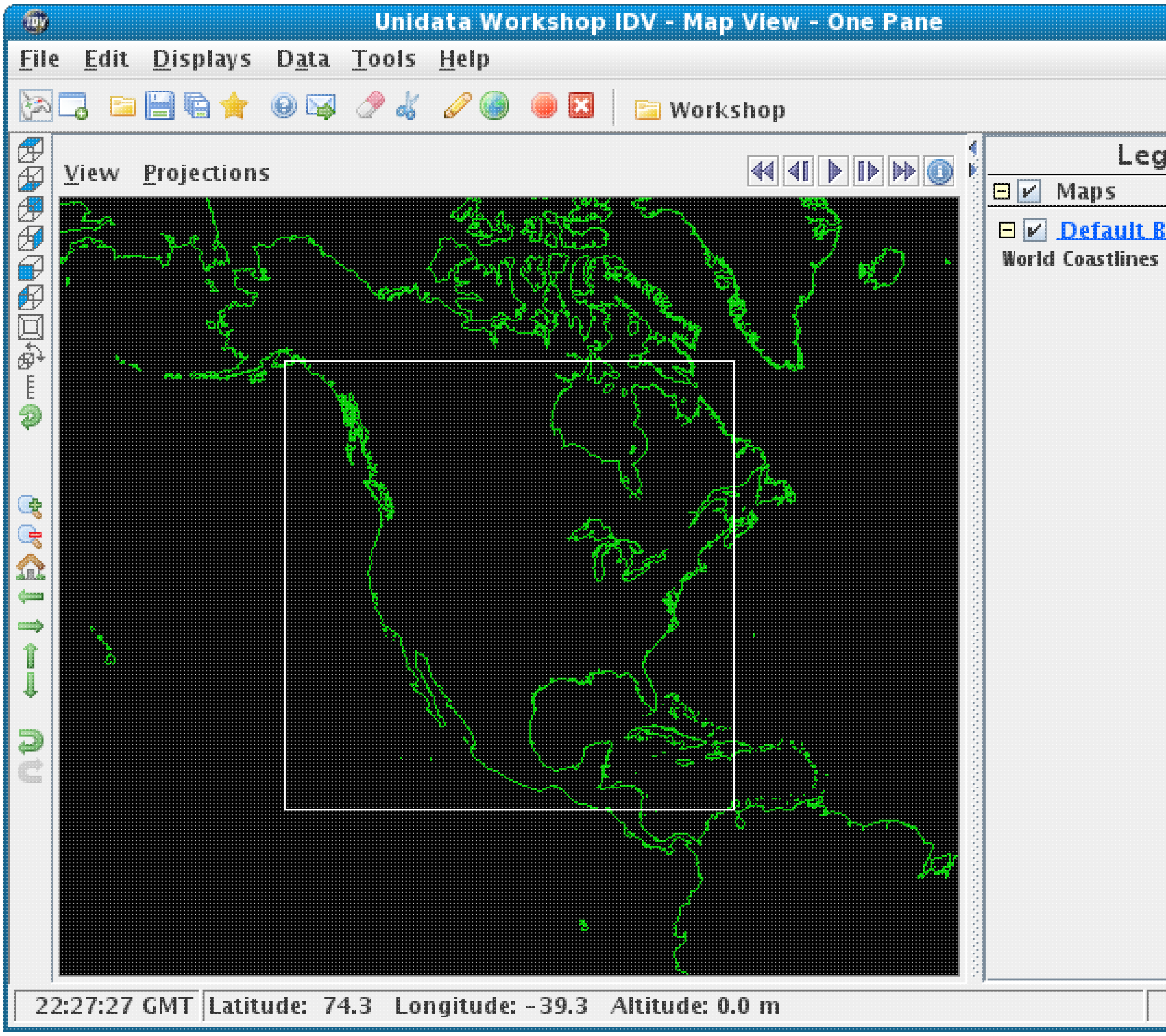
5. When the IDV starts for the first time, you will see 3 windows:
   - `IDV Help Tips` window. An optional window with tips on using the IDV.

For now, close the `IDV Help Tips` window by clicking the `Close` button. We'll look at this in a later exercise.

- `Dashboard` window. Used for selecting data to analyze and visualize.



- `Unidata Workshop IDV` window. This is the main window for displaying data visualizations.



6. Now we are ready to explore the user interface.

Footnotes:

You can cut and paste between your browser and the IDV or a terminal window to save some typing.

- In the browser, highlight the text you want to copy to the IDV by clicking and dragging your mouse over it.
- Press Ctrl-C to copy to the system clipboard
- To paste to the IDV:
    - Click the mouse in the IDV at place you want to paste the text.
    - Press Ctrl-V to paste into the IDV.
- To paste to a terminal window or another application:
    - Click the mouse in the terminal window/application at the place you want to paste the text.
    - Use your terminal/application's operation for pasting (e.g., for X-Windows: click middle mouse button; for many Windows applications: press Ctrl-V)

runIDV script environment

```
java -Xmx${memory}m -jar idv.jar $*
```

- `java` - creates the Java Virtual Machine
- `-Xmx${memory}m` - sets the maximum memory used by Java to ${memory} Mb. You can change this value to increase the memory allotted to the IDV if your system has more memory.
- `-Didv.enableStereo=false` - disables stereo viewing. Set this to true to enable a stereo display.
- `-jar idv.jar` - executes the main class specified in `idv.jar`
- `$*` - allows the user to pass in command line arguments.

2.1 IDV Basic Concepts

In this section, you will learn the basic concepts for using the IDV Graphical User Interface (GUI).

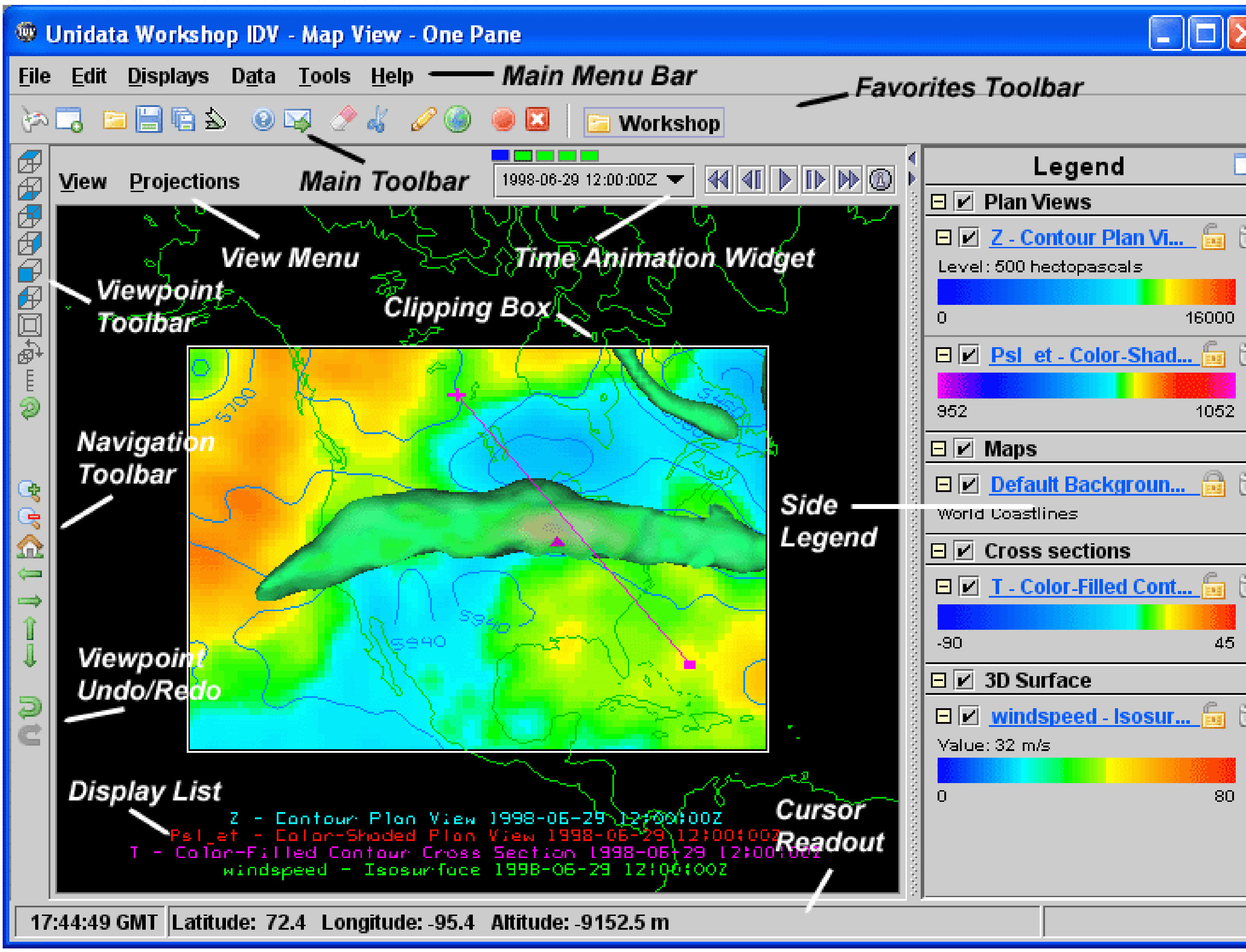
2.1.0 Introduction to the IDV GUI

In this section, we will examine some of the Graphical User Interface (GUI) features of the IDV.
The sections below describe the user interface of the main window of the IDV reference application. First, we'll load in a time sequence of data so we can use some of the menus and widgets in the IDV's GUI and see their effect.

- From the **Favorites Toolbar**, select the **Sample Data Displays** item from the **Workshop** menu.
- When the dialog pops up asking you how you want to load the bundle, click the `OK` button.

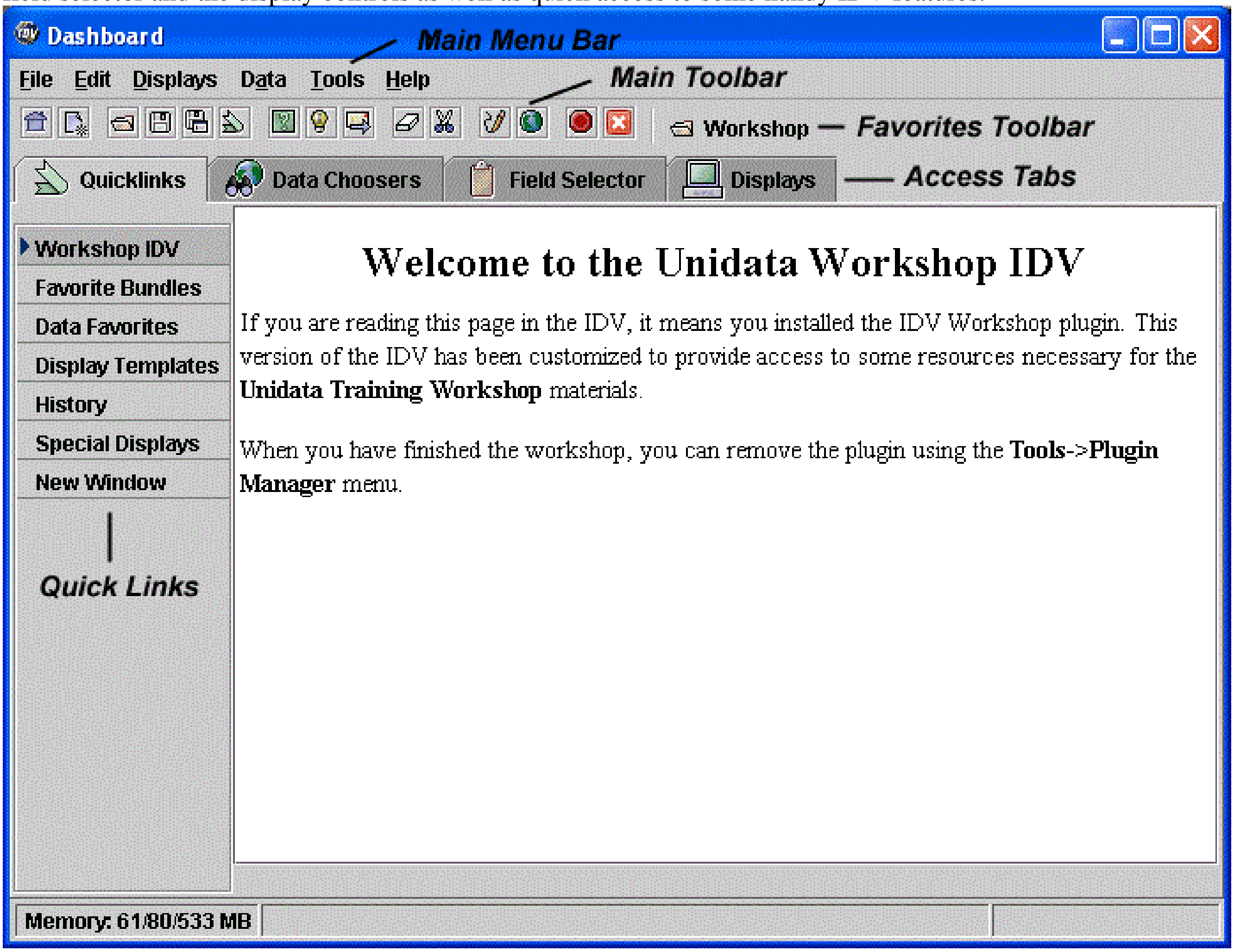
We'll explore the Favorite Bundle mechanism in a later exercise.

View Window
The `View Window` is where the visualizations of the loaded data are displayed. You can have several of these windows as we'll see in a later exercise.



Dashboard

The `Dashboard` window holds many of the non "View" windows. It provides a holder for the data choosers, field selector and the display controls as well as quick access to some handy IDV features.



Notice that the `Dashboard` and `View` windows have identical menu bars. This allows you to control the IDV from either window.

Now let's learn how to navigate around the 3D main window.

2.1.1 Navigating (Zooming, Panning and Rotating) the View Window

The mouse and keyboard can be used to zoom, pan and rotate the view in the `View Window`. (More Info). We'll go through some simple exercises here to get you used to these functions.

1. **Zooming**
   - Zoom in to a portion of the display using the rubber band feature of the IDV. Hold the `Shift` key down on the keyboard. In the `View Window`, click the left mouse button (MB1) and drag the mouse to create a box that covers the region you want to zoom to, and then release the mouse button.
   - To restore the display to the full projection, press the `Ctrl` and `r` keys simultaneously (`Ctrl-r`).
   - You can also zoom by holding the **Shift** key and the right mouse button (MB3) down and move the mouse forward and back. Try zooming this way.
   - Restore the display to its original position with `Ctrl-r`
   - You can also zoom by holding the `Shift` key and pressing the up (zoom in) and down (zoom out) arrows on the keyboard. You have to click in the window first. Try zooming this way.
   - Restore the display to its original position.
   - Finally, if you have a mouse with a scroll wheel, you can zoom out by rolling the wheel forward and zoom in by scrolling the wheel toward you (e.g. bring the view closer). When you are done, restore the display to its original position.
2. **Panning**
   - You can pan the display by holding the `Ctrl` key and the right mouse button (MB3) down and move the mouse around. The display will pan in the direction of the mouse movement. Try this.
   - Restore the display to its original position.
   - You can also pan the display by holding the `Ctrl` key and pressing the arrow keys on the keyboard. Using this method, the display will pan in the direction of the arrow. Try it out. You have to click in the window first.
   - Restore the display to its original position.
3. **Rotating**
   - The main view window is configured as a 3-D display. You can rotate the 3-D view by holding down the right mouse button (MB3) and moving the mouse. Try rotating the display using this method. It takes some getting used to.
   - Restore the display to its original position.
   - You can also rotate the display using the arrow keys on the keyboard. Click in the window and then try using the arrow keys to rotate the 3-D box.
   - Restore the display to its original position.
   - Finally, if you have a scroll wheel on your mouse, can use this to rotate the display. Holding down the `Ctrl` key while scrolling will rotate along the Y axis of the display; holding the `Shift` key will rotate about the X axis. Holding the `Shift+Ctrl` keys at the same time will rotate about the Z axis.` the keyboard. Click in the window and then try using the arrow keys to rotate the 3-D box.
4. **Pulling it all together**

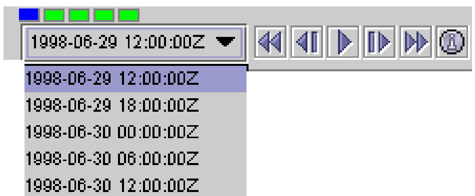
Try different combinations of zooming, panning and rotating. When you are finished, restore the display to its original position.

5. **Moving Selector Points**
    - Some displays have selector points that can be manipulated interactively to change their position.
    - In the sample data display, move the cursor over the endpoint of the red cross section line, click the left mouse button and drag the mouse to move the point.
    - Rotate the display to see the change in the cross section in the main view window
6. You can change the default behavior through the `Navigation` tab of the User Preferences accessed by the **Edit Preferences** menu.

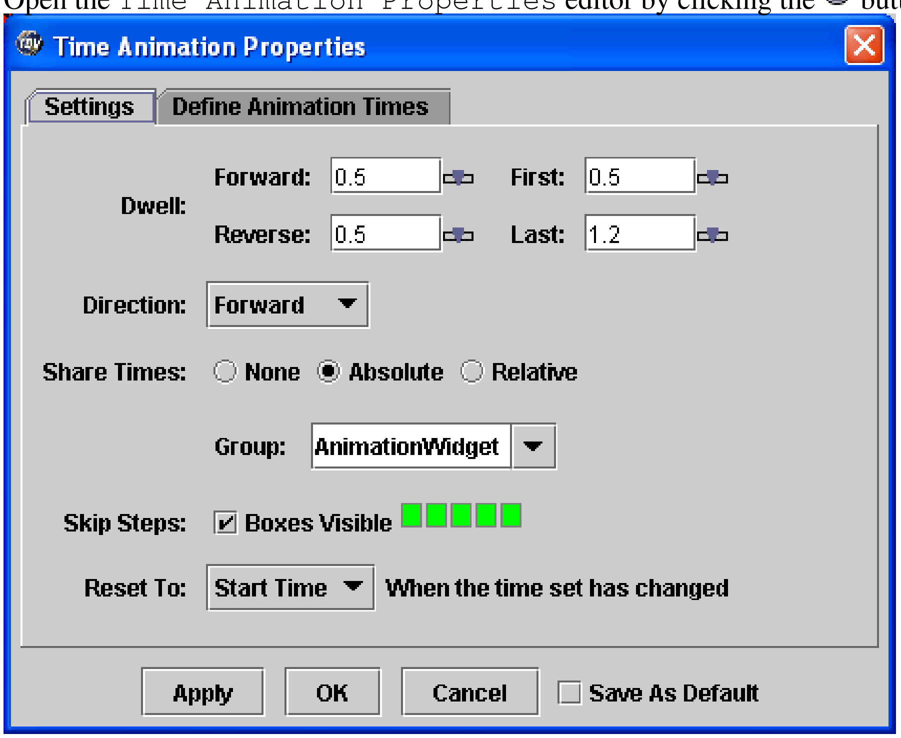
2.1.2 Using the Time Animation Widget

The time animation widget in the IDV can be used to control the properties of a time loop.

1. Click on the dropdown box on the left side of the widget to see the list of times that are in this display. It will look something like this.



2. Select the third time in the list to view the visualization for that time step.
3. Loop through the entire animation by clicking the button on the animation widget.
4. Stop the animation by clicking the button.
5. Return to the first time step by clicking the button.
6. Step forward through the animation one step at a time by repeatedly clicking the button.
7. Step backward through the animation one step at a time by repeatedly clicking the button.
8. Go to the last time step by clicking the button.
9. The row of boxes at the top of the widget is a visual indicator of the current position in the animation sequence. Click on a box to go to that point in the sequence.
10. You can also use the boxes to skip particular timesteps. If you right click (MB3) on a box, it will turn red indicating that that step will be skipped. Right click on a box and then start the animation to see that the timestep is skipped.
11. Open the `Time Animation Properties` editor by clicking the button.



- Change `Direction` to rock back and forth by selecting the `Rocking` option from the dropdown list and then click the `Apply` button.
- Start the animation. When the loop reaches the last time step, the animation steps backward. When it reaches the first time, it goes forward again.
- Set the `First dwell` to be the same value as the `Last dwell`, and then click the `Apply` button to see the effect.
- Set the `Direction` back to `Forward`.

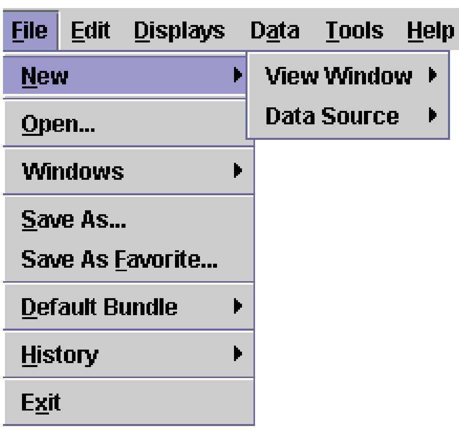
You can also define your own time animation under the `Define Animation Times` tab. We'll look at that in a later exercise.

2.1.3 The Main IDV Menu

In this exercise, we explore some of the items in the main menu bar of the IDV.
The main menu bar is accessible from either the `View Window` or the `Dashboard` window. Detailed information on each item in the main menu bar can be found in the IDV User's Guide.

For this exercise, we will look at the **File** and **Help** menus. The other menus will be explored in subsequent exercises.

The File Menu



The **File** menu is used for opening new `View Windows`, restoring the `Dashboard`, importing and exporting bundles and exiting the IDV. As a simple exercise, we will close the `Dashboard` and then open a new one, since this is a common function.

1. Close the `Dashboard` window by clicking on the **X** in the title bar of the window.
2. Open a new `Dashboard` by selecting the **File New View Window Windows Dashboard** menu item.
3. You can have multiple view windows open at the same time. Open a new `View Window` by selecting the **File New View Window Map View One Pane** menu item. Open a two pane Globe Display.
4. Close these new `View Windows` by clicking the **X** in the upper right corner.

We'll look at other portions of the **File** menu in subsequent exercises.

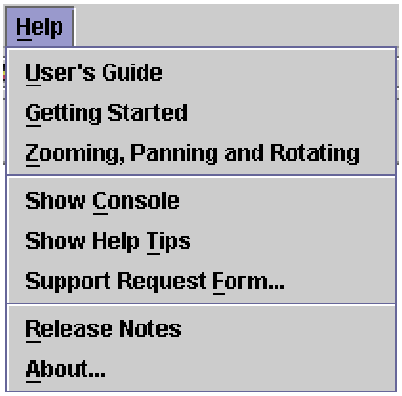
The Edit Menu

The **Edit** menu can be used to clear out displays and data sources as well as access the formula editor and user preferences.

1. First, let's clear out all the displays we have loaded. Select the **Edit Remove All Displays** menu. All the displays except for the map should be removed from the view.
2. In the `Field Selector`, notice that the data source is still listed. We could create new displays from that data source if we wanted, but lets remove it. Select the **Edit Remove All Displays and Data**.

The Help Menu

The **Help** menu is used to access the incorporated documentation and other features that provide information about the IDV.



1. First, let's look at the User's Guide. Select the **Help** **User's Guide** menu item to bring up the IDV User's Guide.
2. You can select any topic in the navigation pane at the left side of the viewer. For example, expand the `Appendix` tab, then the `Examples of Display Types` tab and select the `Flow Displays` entry in the list to view that page.
3. Now, use the **Help** **Zooming, Panning and Rotating** menu item in the main menu bar of the IDV to go to the page that provides an overview of mouse and keyboard use.
4. Close the `IDV User's Guide` window by clicking on the **X** in title bar of the window.
5. Now, let's bring the `Help Tips` window back up by selecting the **Help** **Show Help Tips** menu item. The help tips in this window are designed to give you some tips on using the IDV and quick access to sections of the User's Guide. Click the `Next` button until you get to the **Zooming, Panning, and Rotating** tip. Click on the link to bring up the User's Guide section on this topic.
6. Uncheck the `Show tips on startup` checkbox and then click the `Close` button. The next time you start the IDV, the `Help Tips` window will not automatically pop up.
7. There are several items in the **Help** menu that are useful if you are having problems and need to contact Unidata User Support. The **Help** **Show Console** menu item brings up the IDV console where all error messages are shown. Another important menu is the **Help** **About...** menu which lists the version of the IDV that you are running. You should always indicate the version when you contact Unidata User Support with any problems you may have. Lastly, if you are having problems with the IDV, the **Help** **Support Request Form...** menu allows you to submit a problem report directly to Unidata. This should be used as a last resort after you have followed all the recommended steps in the User's Guide Documentation and Support. Be sure to fill in all the fields and most importantly to provide a detailed description of the problem.

Now, let's look at the **View** and **Projections** menus in the `View Window`.

Footnotes:
The items in the top part of the `Help` menu are shortcuts into relevant sections of the User's Guide.

2.1.4 The View and Projections Menus

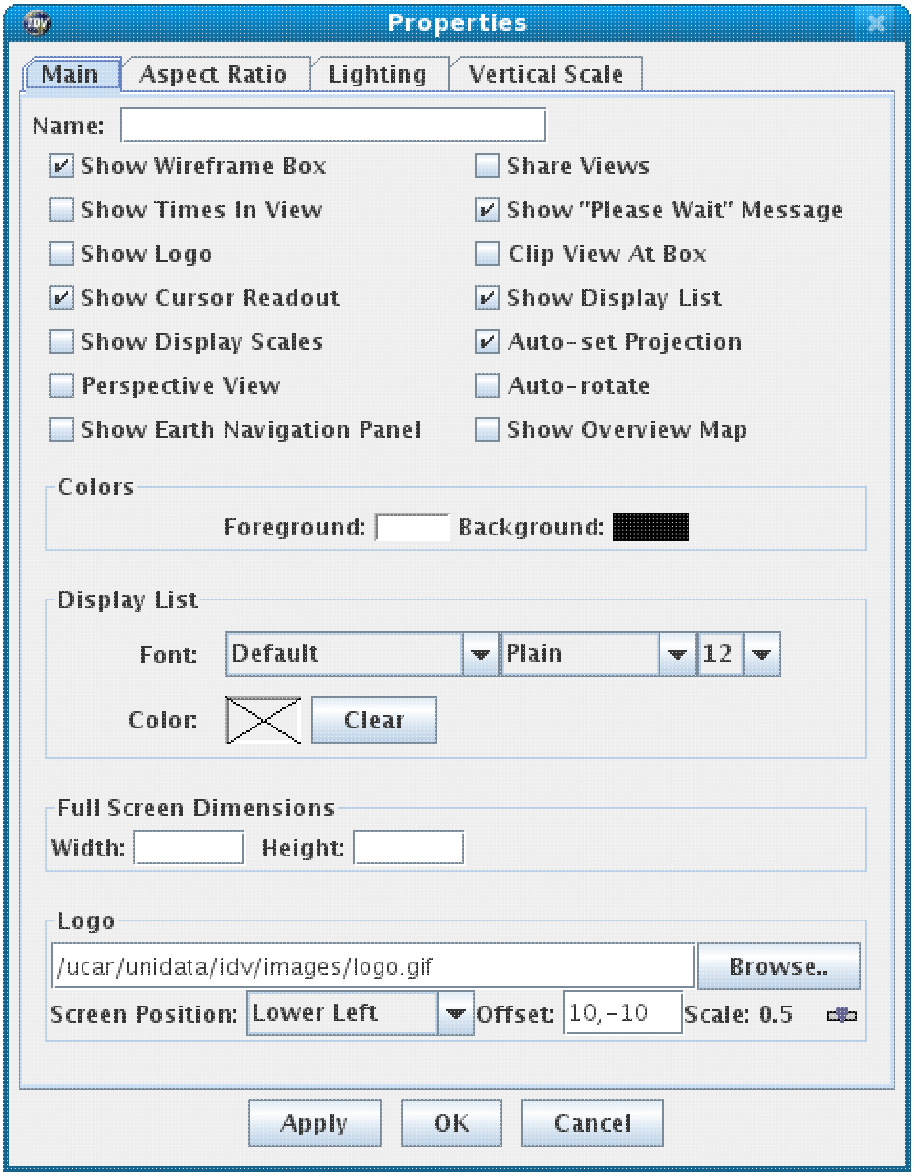
In addition to the main menu bar, each `View Window` has another menu with options for controlling just that view. In this section, we will explore some of the features of the **View** and **Projections** menus associated with each `View Window`.

The View Menu

Each `View Window` also has a **View** menu that lets you change the point of view, capture the state of the display, and make other changes in the overall view.

First let's look at the Properties dialog for the View Window:

1. Open the `View Window Properties` dialog with the **View** **Properties** menu.



2. Uncheck the `Show Wireframe Box` item to turn off the 3D box in the display then click the `Apply` button. Check it again to turn the box back on and click the `Apply` button.
3. Uncheck the `Show Cursor Readout` item to turn off the lat/lon/alt readout at the bottom of the `View Window` and click the `Apply` button.
4. Uncheck the `Clip View At Box` menu item to turn off clipping of the map lines and other displays at the 3D box and click `Apply`. Check it again to turn on clipping.
5. Check the `Show Times In Display` item to turn on the time label in the view window. Uncheck the `Show Display List` item and click `Apply`. The Display List is no longer shown, but the time label is now shown. Note that the time is also shown as part of the Display List for each of the displays and in the `Time Animation Widget`.
6. Change the foreground and background colors by clicking the `Color` widgets and selecting a new color. After making your selection, click `Apply`.

The default settings for each of these can be set through the User Preferences which will be covered later in the workshop.

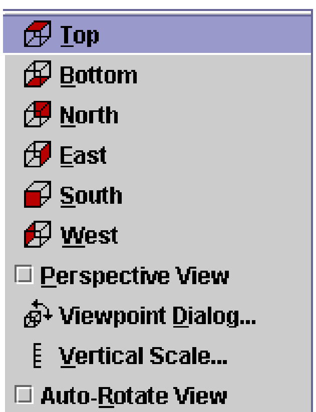
Other View Menus

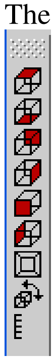
Some of the properties that we just changed can also be set through other **View** menus.

1. Select the **View** **Show** menu to see which of the items we changed above are available. Use some of these to change your display.
2. Select the **View** **Color** menu to change the foreground and background colors of the display. You can select one of the pre-defined foreground/background color pairs, or create your own. Select one of the predefined options to change the display colors. Select **View** **Color** **White on black** menu to return to the system default.
3. Select the **View** **Full Screen** menu to expand the view to the full screen. Click the X to return to normal screen mode.

The View->Viewpoint Menu

Another useful menu item is the **Viewpoint** menu. This menu can be used to control which side of the 3D box you are looking at.



1. The five top menu items, **Top** through **West** instantly reset the point of view to one of the orthogonal directions. The menu icons show which side of the box will be displayed. Note that **Top** is the same as the reset command **Ctrl-r** which we used earlier.
2. Select the **South** menu then rotate the box with the keyboard or mouse to see the orientation.
3. Reset the view to the top.
4. The **Perspective View** item toggles between a parallel (unchecked) and perspective (checked) view. Check the box and see what effect it has.
5. The `Viewpoint toolbar`:
   

provides the same functionality as the menu items described above.

Other Viewpoint menu items

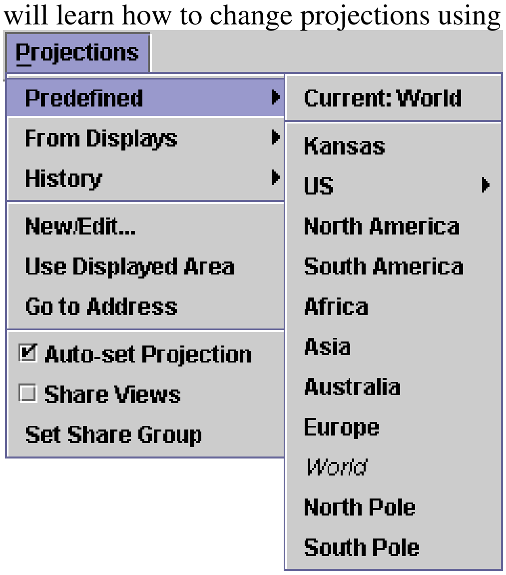
Viewpoint Dialog... displays a dialog box for you to enter the azimuth and tilt of a desired point of view.

Vertical Scale...displays a dialog box to set the linear vertical scaling between the top and bottom the wireframe box. The default is 0 to 16 km. This can also be changed through the `Vertical Scale` tab of the `View Window Properties` dialog.

The Projections Menu

The `View Window`'s **Projections** menu let you change and define the map projection.

Changing Map Projections
Map projections define the region of the earth that you are viewing. The IDV is configured with a set of pre-defined map projections. In a later exercise, you will learn how to create your own. In this exercise, we will learn how to change projections using the system set.



1. Select the **Projections** **Predefined** **US** **CONUS** menu item to switch to the continental US projection.

2. Choose another projection from the **Maps Projections Predefined** menu

The **Projections From Displays** menu has a list of projections from the data of any displays.

The **Projections History** menu has a list of recently used projections.

The **Projections New/Edit..** menu item is used to bring up the `Projection Manager` which is used to define projections. You will use this in a later exercise to define your own projections.

Other Projections Menu Items

- The **Auto-set Projection** menu item allows you to control whether the display projection is changed when you load in new data.
- The **Share Views** menu item allows you to share the maps and projections between different `View Windows`.
    1. Select the **File New View Window Map Display One Pane** menu item to bring up a new `View Window`.
    2. Rotate the view in each display.
    3. Use the **Projections Predefined** menu in each `View Window` to change the projections to be different in each display.
    4. Check the **Projections Share Views** item in each `View Window`.
    5. Rotate the view in one display. Notice how it affects the other.
    6. Change the projection in one of the displays. The same projection will be applied to the other display.
    7. Close the second `View Window` by clicking the **X** in the upper right corner of the window.

2.1.5 Changing Background Maps

The IDV is distributed with a set of maps of varying resolution and coverage. You can change the maps being displayed in any `View Window` using the `Map Display Control`.
By default, only the world outlines are displayed when you first start the IDV. You can change the default to use other system maps, or add in your own. The IDV can use McIDAS map files and ESRI shapefiles for map backgrounds.

1. Click on the `Default Background Maps` link in the `Side Legend` panel to bring up the control window in the `Dashboard`.
2. For each map in the widget, you can control the visibility, the line width, the line type, and color.
    - Change the color of the `North & Central America` map to green and set it visible.
    - Set the width of the `World Coastlines` map to 3 pixels wide.
    - Set the width of the `World Coastlines` map back to 1 pixel wide.
3. When you are finished, save this as the default settings using the **File Default Maps Save as the Default Map Preference** menu in the `Default Background Maps` control.

2.1.6 Projection Manager

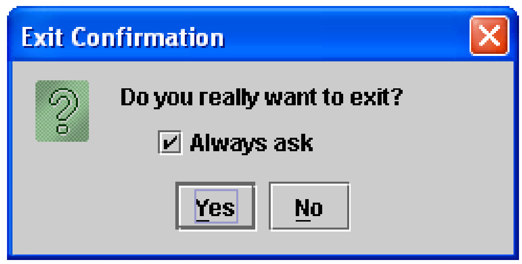
In this section we will cover how to create and edit map projections.

1. Select the **Projections New/Edit** menu item.
2. View an existing projection.
    - Select "US>CONUS" and press `Edit`
    - Right mouse/drag in map to scroll. Shift+Left mouse drag to zoom.
    - Press `Cancel`
3. Create a new projection.
    - Click on the `New` button.
    - In the `Define/Edit Projection` window enter the name "Workshop>Colorado" in the `Name` box.
    - Leave the `Type` as "LatLon".
    - Click in the map view on the left and delete the existing bounding box by pressing the `Delete` key.
    - Click in the **Maps** menu and turn on the `World Coastlines` map.
    - Zoom in over the United States by holding the Shift key down and dragging the mouse with the left mouse button down.
    - Draw a bounding box over Colorado by dragging with the left mouse button pressed. You can adjust the size of the box by dragging on the boxes on the box. You can move the box by clicking and dragging on the box lines. When you are finished, click the `Save` button.
    - Back in the **Projections Predefined** menu select **Workshop Colorado** to view the projection.
4. Set the default projection to "US>CONUS"
    - Open the Preferences Manager from the **Edit Preferences** menu.
    - Select the `Main View` tab.
    - Select the `US>CONUS` projection in the `Default Projection` dropdown list.
    - Click the OK button.
    - The next time you start the IDV, that will be your default projection.

2.1.7 Exiting the IDV

Exiting the application is easy.
For now, we'll exit out of the IDV in preparation for the next exercise.

1. To exit the IDV application, select the **File->Exit** menu item or click the  icon in the main toolbar.
2. You be given a chance to cancel this action if you hit it inadvertently.



3. Click `Yes` to exit. If you would like to bypass this message in the future, uncheck the `Always ask` option before clicking `Yes`

3 Accessing and Displaying Data

In this section, we will discuss the data selection process and how to load in most of the types of data that the IDV supports.

3.0 Data Selection Overview

In the IDV, the process of analyzing and visualizing data takes the following steps:

1. Loading in a `Data Source`.
2. Selecting the parameter or diagnostic formula to use for the display.
3. Selecting the type of display (and optionally selecting a range of times, levels and spatial subsets).
4. Creating the display.

In the following sections, we will examine this process in more detail.

3.1 Using the Data Source Chooser

The Data Source Chooser is used to load data sets into the IDV.

Data Source Chooser Window

The Data Source Chooser has tabs for selecting the different <u>types</u> of data that the IDV can display. Let's examine the features of this pane.

1. Start the IDV
2. Open the `Data Source Chooser`.
3. There is a tab for loading in each of the types of data that the IDV supports. Click through each of the tabs to see what is there. We will examine each tab in more detail in the subsequent exercises.

Footnotes:

Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.2 Working with Gridded Data

Gridded data can be the output from a numerical model or an objective analysis of observational data. Each gridded data source is comprised of a set of 2- and/or 3-D fields. Each field may have of one or more times associated with it. The IDV supports a variety of ways of visualizing gridded data. In this section we will explore some of these.

3.2.0 Loading Gridded Data

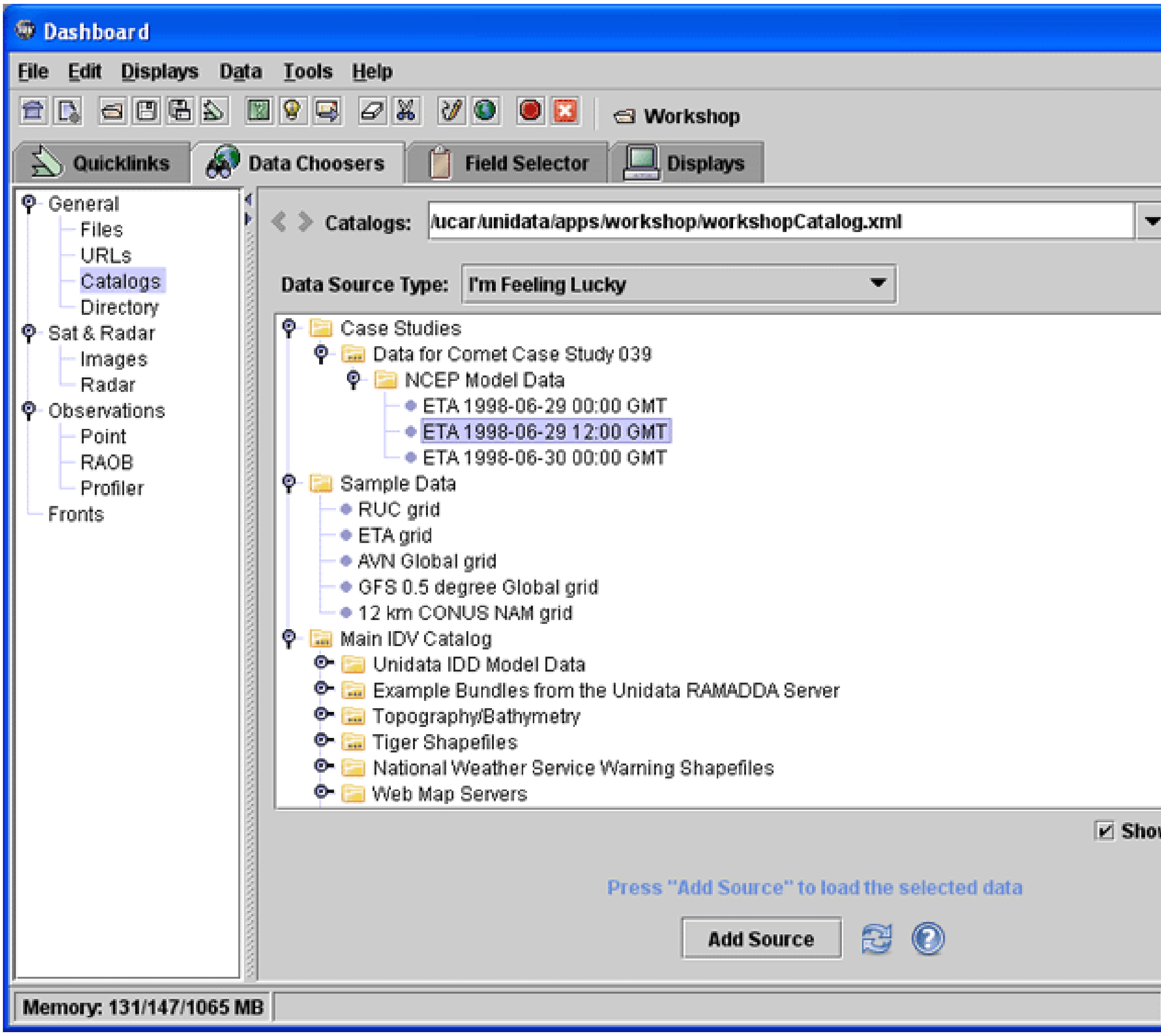
Gridded data can be accessed in the IDV from a remote server or from local files. For the exercises in this section, we will use ETA model output from the COMET Case Study on Bow Echoes (39) from June 29, 1998.

1. Start the IDV
2. Load in the 12 Z ETA model run from June 29, 1998.
   - Open the `Data Source Chooser` window.
   - Click on the `Catalogs` tab.
   - From the catalog selector drop-down, select the IDV workshop catalog that is included in the plugin:

     /ucar/unidata/apps/workshop/workshopCatalog.xml

     If it is not in the list, type it into the selector area and press the `Update` button. (Cut and Paste tip)

     A tree view of the data catalog will be displayed in the window below the selector.
   - Expand the `Data for Comet Case Study 039` tab, and then the `NCEP Model Data` tab.
   - Select the `ETA 1998-06-29 12:00 GMT` entry, and then press the `Add Source` button.



Footnotes:
You can cut and paste between your browser and the IDV or a terminal window to save some typing.

- In the browser, highlight the text you want to copy to the IDV by clicking and dragging your mouse over it.
- Press Ctrl-C to copy to the system clipboard
- To paste to the IDV:
    - Click the mouse in the IDV at place you want to paste the text.
    - Press Ctrl-V to paste into the IDV.
- To paste to a terminal window or another application:
    - Click the mouse in the terminal window/application at the place you want to paste the text.
    - Use your terminal/application's operation for pasting (e.g., for X-Windows: click middle mouse button; for many Windows applications: press Ctrl-V)

Open the `Data Source Properties` editor by either:

- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

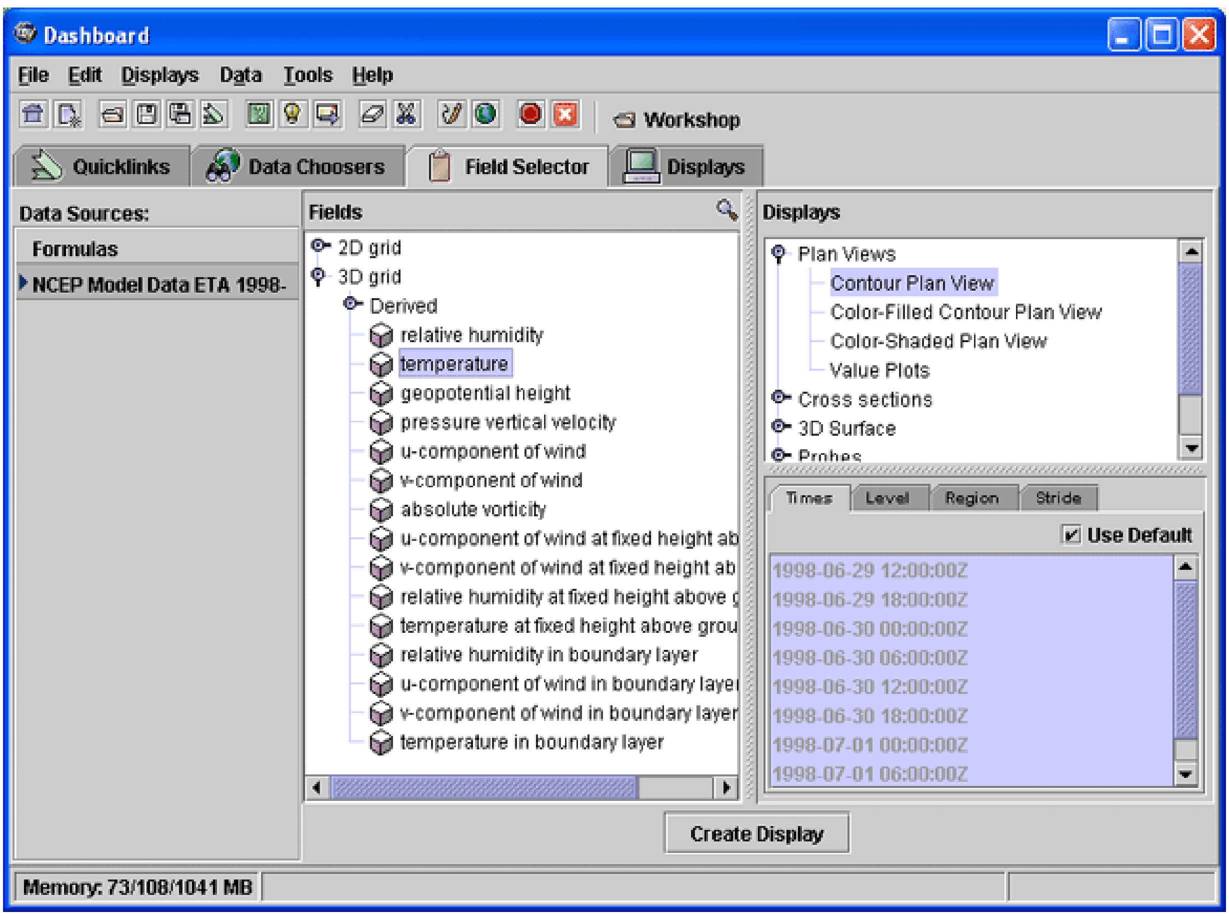
3.2.1 Using the Field Selector

The Field Selector is used to select data sources, parameters, data times, and types of displays to use for creating visualizations.

Field Selector

The `Field Selector` consists of 4 panels:

- `Data Sources` - used for selecting data sources that are currently loaded into the IDV. `Formulas` is a particular data source for applying formulas to other data sources.
- `Fields` - displays the list of data choices (parameters, formulas) for the selected data source.
- `Displays` - holds the list of display types applicable to the selected field.
- `Data Subset` - depending on the type of data, this panel displays the list of times, levels and spatial subsetting associated with the selected data choice.



Data Sources

When the IDV reads in a data source, most of the time it only reads enough metadata to be able to create a list of `data choices`. Some of the datasets may be quite large and this allows you to only read those portions that you need instead of the entire dataset.

1. Right click the `NCEP Model Data ETA` data source in the `Data Sources` panel of the `Field Selector` window and select the **Properties** menu.
2. Each `Data Source Properties Dialog` will have one or more tabs for setting and viewing the properties of that data source. Tab through the properties of this data source.
3. Click the `Cancel` button to close the window

Data Choices (Fields)

When a data source is loaded, a set of `data choices` are created from the metadata. This allows you to select a subset of the dataset.

1. Select the `NCEP Model Data ETA` data source in the `Data Sources` panel of the `Field Selector` window.
2. In the `Fields` panel, you will see two tabs - `3D grid` and `2D grid` indicating that this dataset has both 2D and 3D fields in it.
3. Click on the little tab to the left of the `3D grid` field to expand that tab. The list contains those 3D fields from the dataset that the IDV can display.
4. Select the `temperature` field. When you select a field, the `Displays` and `Data Subset` panels are populated with entries applicable to that field.

Derived Choices

When a data source creates the list of data choices from the parameters in a dataset, it also uses a library of pre-defined derived quantities to create a list of choices for fields that can be calculated from the fields in the dataset.

- Expand the `Derived` tab.
- In the expanded list, select `Dewpoint (from T & RH)`. This dataset does not have a dewpoint field in it, but it does have temperature (T) and relative humidity (RH) which can be used to calculate dewpoint.

Displays

When each data choice is created, the metadata for that choice is used to determine which displays are applicable to that field.

1. Look at the list of available display types for your Dewpoint selection in the `Displays` panel.
2. Select the `Flow Vectors (from GridRelative_u & GridRelative_v)` field and notice that the list of displays changes.
3. Select the `Sounding Data (T & dewpoint only)` field and notice that you get another list of displays.
4. Select `Dewpoint (from T & RH)` again.
5. In the `Displays` panel, select `Contour Plan View` so we can create a horizontal slice of the data and display it as contours.

Data Subsetting

The `Data Subset` panel allows you to temporally and spatially subset the data you've chosen. For this exercise, we will look at subsetting by time and level.

Times

Each data choice may or may not have times associated with it which are displayed in the `Times` panel. If there is a set of times, you can select all or just a subset of times to be displayed. There are two ways you can select a subset of times.

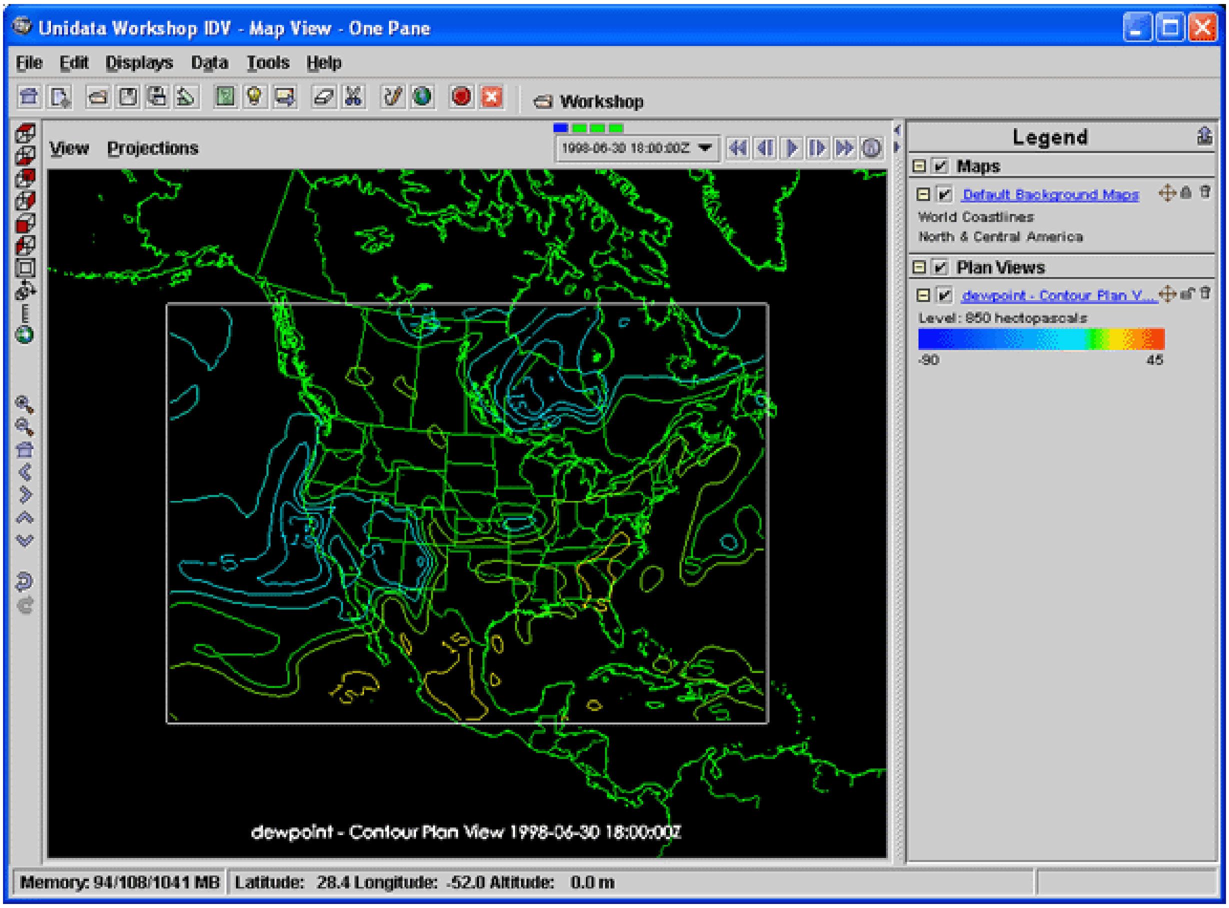
- You can set the time subset for all parameters in the dataset.
    1. Double click (or right click and select the **Properties** menu) on the dataset name in the `Data Sources` panel to bring up the `Data Source Properties` editor.
    2. This dataset has ten time steps. In the `Times` tab, uncheck the `Use Default` checkbox. You can now select the subset of times. Select the first five (5) times in the list (`1998-06-29 12:00:00Z` to `1998-06-30 12:00:00Z`).
    3. Click the `OK` button to apply this change and close the window.
    4. Notice that the change is reflected in the `Times` tab of the `Data Subset` panel when you select `Dewpoint` this time.
- You can set the time subset for a single parameter in the `Times` tab.
    1. In the `Times` tab of the `Data Subset` panel, uncheck the `Use Default` checkbox.
    2. In the list of times in that panel, select the last four (4) times in the list (`1998-06-30 18:00:00Z` to `1998-07-01 12:00:00Z`).

Levels

Each data choice may or may not have levels associated with it which are displayed in the `Level` tab of the `Data Subset` panel. If you do not select a level, the first level is read in unless you are creating a display that needs a volume of data.

- You can set the level subset for a single parameter in the `Level` panel.
    1. In the `Level` tab, select the `850 hectopascal` level.

When you are finished making all your selections, click the `Create Display` button to apply them and create the display. The subset of data you selected will be read from the data source and displayed in the currently active `View Window`.

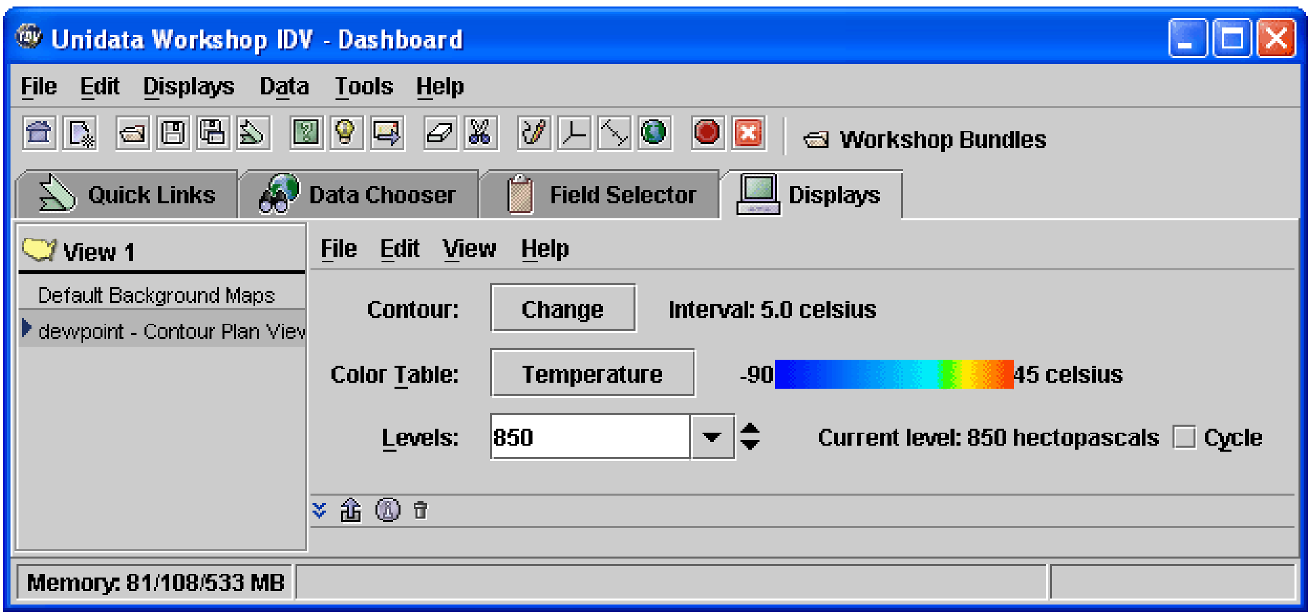

Unidata Workshop IDV - Map View - One Pane
File Edit Displays Data Tools Help
Workshop
View Projections
1998-06-30 18:00:00Z
Legend
Maps
Default Background Maps
World Coastlines
North & Central America
Plan Views
dewpoint - Contour Plan V...
Level: 850 hectopascals
-90
45
dewpoint - Contour Plan View 1998-06-30 18:00:00Z
Memory: 94/108/1041 MB
Latitude: 28.4 Longitude: -52.0 Altitude: 0.0 m

3.2.2 Using the Display Controls

Each display has a **display control** associated with it. A `Display Control Window` is a panel or window with menus, buttons, sliders, and other widgets to control a display. When a display is first created its **display control** is added to the `Dashboard` or pops up in a separate window depending on your configuration. The control also has a legend shown in the `View Window`'s legend panel.

The Display Control Window

The control window is used to change the properties of the display. Each display type may have a different set of widgets in the control window. For our dewpoint display, the control looks like this:



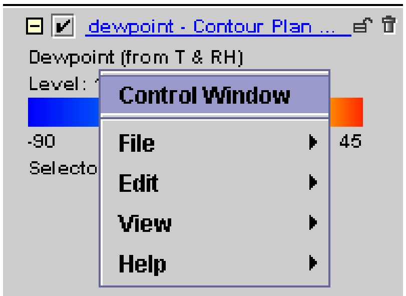
The functionality of each display control will be covered in subsequent exercises so we won't go over them here.

Display Control Menus

In addition to the widgets, each display control has a set of menus associated with it that allow you to change the behavior of the control, change some of the display characteristics, or get help on using the control.

The menus can be accessed from the menu bar in the control window:

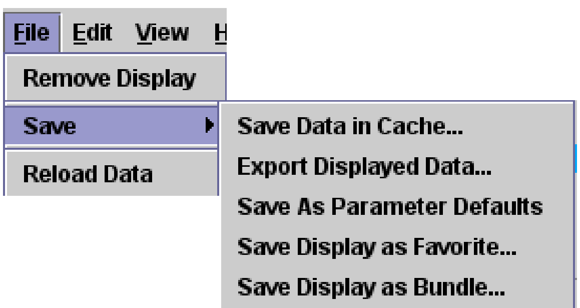
or by right clicking on the legend label:



Let's briefly look at what is in each of these.

File Menu

The **File** menu allows to remove, save or reload the this control. Each display control may have a different set of entries under this menu, depending on the type of data being displayed.

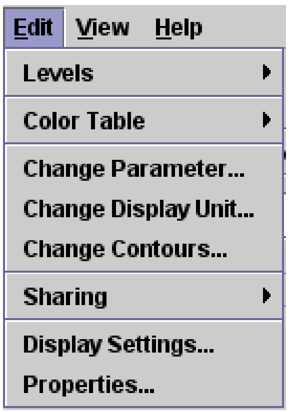


Some common choices are:

- **Remove Display**
  Use this item to remove the display from the `View Window`. You can also remove the display by clicking on the trash can icon () in the side legend or at the bottom of the control in the `Dashboard`.
- **`Save Save Data in Cache...`**
  This menu item allows you to save the currently loaded field in a data cache. This is useful for saving the results of a complicated formula for future displays or even for use in another formula.
- **`Save Export Displayed Data...`**
  Some controls allow you to export the displayed data to a disk file. Depending on the control, the output could be a netCDF or a comma-separated value (CSV) file.
- **`Save Save as Parameter Defaults`**
  Allows you to save current properties (e.g., color table, contour information, display unit) of this parameter as the default for other displays of this parameter.
- **`Save Save Display as Favorite...`**
  Allows you to save this display as a special template that can be applied to other parameters of a similar type.
- **`Save Save Display as Bundle...`**
  This menu item allows you to save the display as a bundle that you can then associate with a new data source.
- **Reload Data**
  This menu item allows you to reload the data in the control. It is useful if the underlying data source has changed.

Edit Menu

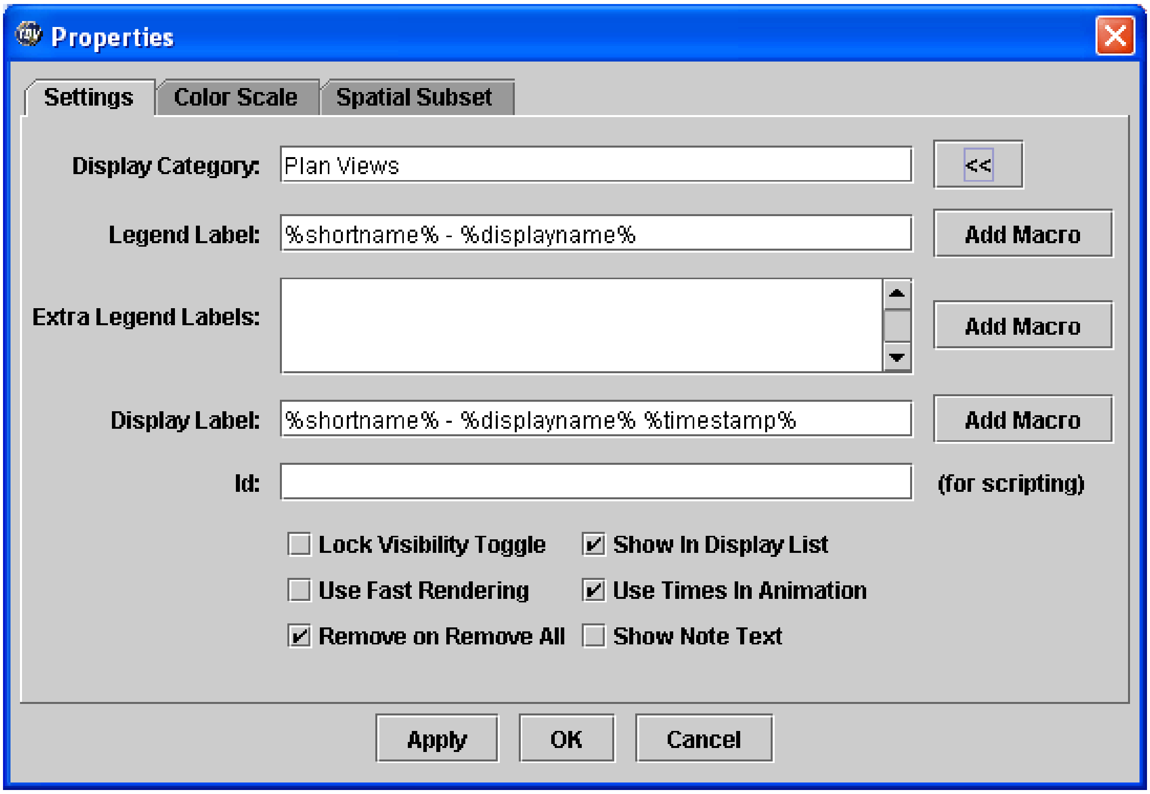
The **Edit** menu allows you to edit some of the properties of the display. Each display control may have a different set of entries (or none) under this menu, depending on the type of data being displayed.



Some common choices are:

- **Levels**
  For displays that have a multiple vertical levels, this menu allows you to select a particular level.
- **Color Table**
  For displays that have a color table this menu allows you to edit the color table, change the range, choose a new color table, etc.
- **Change Parameter...**
  Some displays allow you to change the data parameter that is being used
- **Change Display Unit...**
  Some displays allow you to change the unit that is used in the display, e.g., changing temperature plots in Kelvin to Celsius.
- **`Sharing Sharing On`**
  Use this checkbox menu item to share some of the properties of this display with similar display types in the same share group.
- **Display Settings...**
  Use this menu to bring up the `Display Settings` dialog. You can use that to save some of the controls settings as a favorite Display Setting or apply them to other controls in the view window. The use of this dialog is covered in a subsequent exercise.
- **Properties**
  Use this menu to bring up the `Display Control Properties` editor.

Display Control Properties



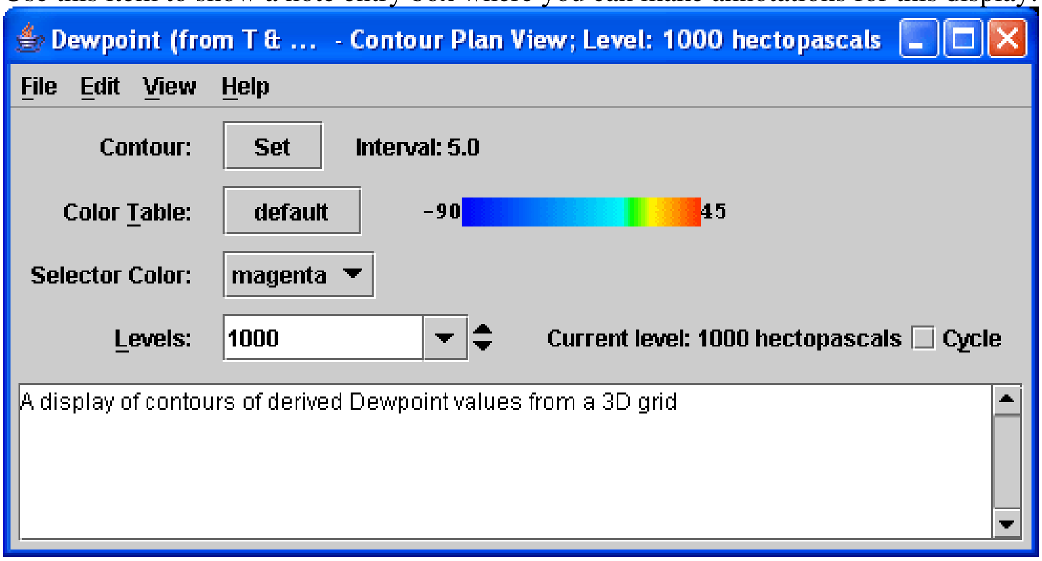
The `Display Control Properties` dialog allows you to change some of the features of the display and how the control works:

- **Display Category**
  You can change the display category for this control. You can select from existing categories, or type in a new one. The display category, is used to group displays within the side legend.
- **Legend Label**
  You can change the label that describes this display and is shown in the legend. You can add in plain text or insert some pre-defined *macros* using the `Add Macro` button. You can also use this button to save the configuration for other displays of this type.
- **Extra Legend Labels**
  You can add extra labels to the legend by entering them here.
- **Display Label**
  You can change the formatting of the label in the main display similar to changing the Legend Label.

You can use the checkboxes in the dialog to change other properties. Some common choices are:

- **Lock Visibility Toggle**
  This locks/unlocks the toggling of visibility for this display. .
- **Use Fast Rendering**
  This option will render data without checking for projection seams. It increases the rendering speed at the expense of accuracy.
- **Remove on Remove All**
  This checkbox lets you control when a display is removed if the user selects Remove All Displays. For example, the default map display has this turned off.
- **Show in Display List**
  This checkbox lets you control whether a display is shown in the display list.

- **Use Times in Animation**
  If this display has times associated with it, this option tells the IDV whether they should be used in setting the animation steps.
- **Show Note Text**
  Use this item to show a note entry box where you can make annotations for this display:

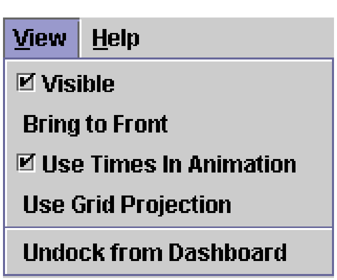


Try these changes:

1. **Change the display's category** - Bring up the `Display Control Properties` dialog. Change the category to "Derived Parameters" and then click the `OK` button.
2. **Change the name of the display** - Right click on the name and select the **Edit Properties** menu. In the text field, type "My cool dewpoint display" then click the `OK` button. Try different combinations of the macros to see their effect.
3. **Change the display label** - Bring up the `Display Control Properties` dialog. Highlight and delete the `%displayname%` macro in the Display Label field. Click the `Add Macro` button and select the **Level** macro. Click the `OK` button and see the change in the display.
4. **Add a color scale** - Bring up the `Display Control Properties` dialog. Select the `Color Scale` tab in the dialog. Check the `Visible` checkbox, set the `Position` to **Left**, and set the `Label Color` to white, then click the `OK` button and see the change in the display.

View Menu

The **View** menu allows you to change the way the view works. Each display control may have a different set of entries under this menu, depending on the type of data being displayed.



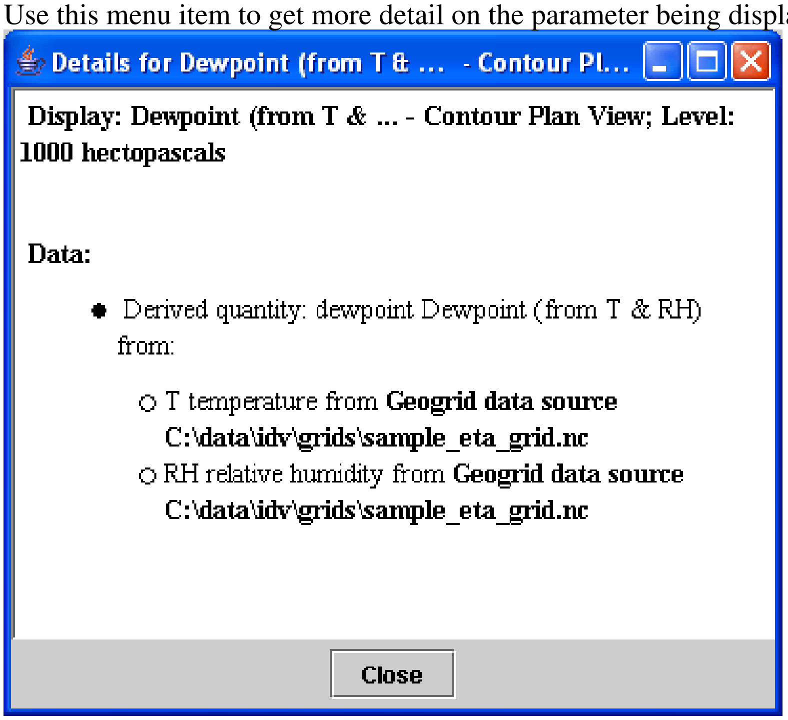
Some of the things you can do with this menu are:

- **Visible**
  Use this checkbox menu item to toggle the visibility of the display in the `View Window`.
- **Use Times in Animation**
  If this display has times associated with it, this option tells the IDV whether they should be used in setting the animation steps.
- **Bring to Front**
  This will reorder the displays to put this in the foreground of other displays at the same vertical level. It is useful when displays have transparent color tables.
- **Use Data Projection**
  This menu item sets the projection in the currently active view window to that of the data used by this control.
- **Dock in/Undock from Dashboard**
  This menu item allows you to dock or undock the display control window in/from the `Dashboard`.

Help Menu

The **Help** menu allows you to get more information on using the control and what data is being displayed.

With this menu you can:

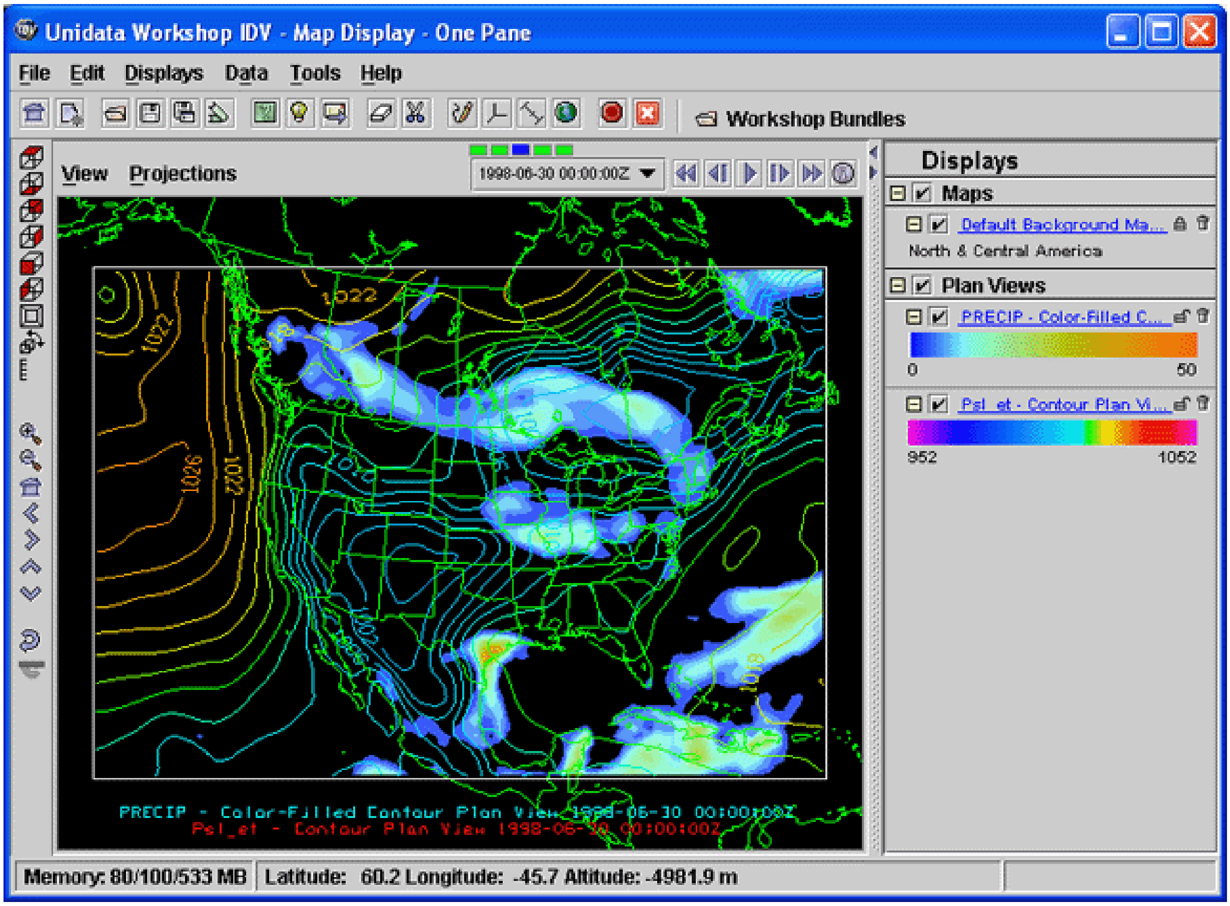
- **Details...**
  Use this menu item to get more detail on the parameter being displayed:


- **User's Guide**
  Use this menu item to access the IDV User's Guide section pertaining to the particular display control.

3.2.3 Plan Views of Gridded Data

Plan views are horizontal displays of two dimensional data. These data can be true 2D fields (e.g., sea level pressure) or a slice of a 3D dataset at a constant pressure or height (e.g., 500 hPa heights). The IDV can display these plan views as contours, color filled contours, or color shaded displays.

1. First, let's create a display of contours of sea level pressure.
   - If you have not already done so, use the **Edit Remove All Displays** menu to remove the displays created in the previous exercise. Set the viewpoint to the top.
   - In the `Field Selector`'s `Data Sources` panel, select the `NCEP Model Data ETA data source` that you loaded in during the previous exercise.
   - In the `Fields` panel, expand the `2D grid` tab. Select the `mean sea level pressure (ETA model reduction)` field.
   - In the `Displays` panel, select `Contour Plan View` then click the `Create Display` button. The data will be read from the remote server and displayed in the main map window of the IDV.
   - From the control window for this display, you can change the contour interval. Click the `Change` button next to the **Contour:** label to open the `Contour Properties Editor`. In the editor, change the contour interval to **2** and click the `OK` button.
2. Now, let's add a display of color-filled contours of precipitation.
   - In the `Field Selector`'s `Fields` panel, select the `total precipitation over accumulation interval` field.
   - In the `Displays` panel of the `Field Selector`, select `Color-Filled Contour Plan View` and then click the `Create Display` button.
   - Since this is a display of precip over an accumulation interval, there will be no display for the first time period of the model. To see the display for subsequent times, start the animation.
   - Stop the animation and return the display back to the first time in the sequence.

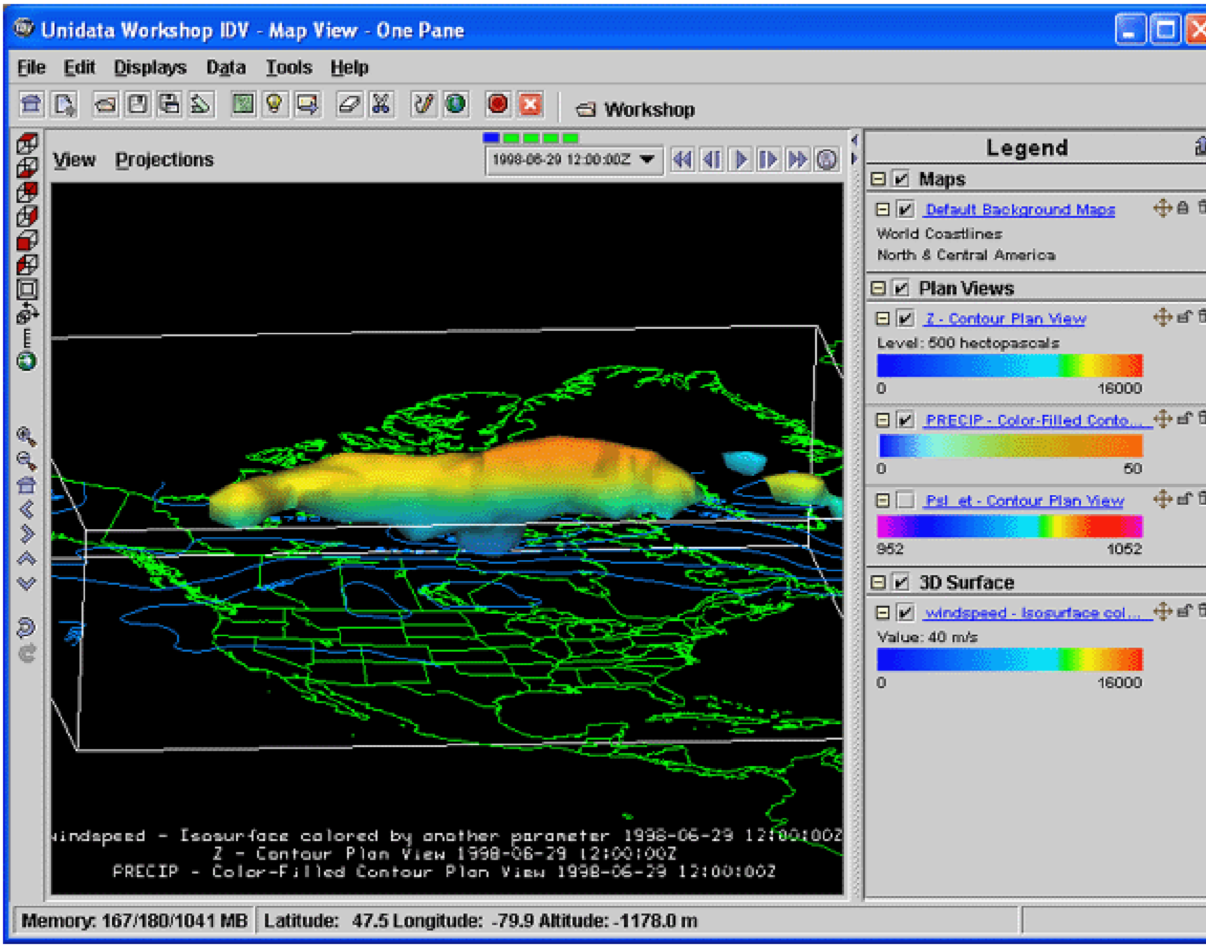


3. Overlay contours of 500 hPa geopotential height on the same display.
   - In the `Fields` panel, expand the `3D grid` tab. Select the `geopotential height` field.
   - In the `Displays` panel, select `Contour Plan View`.
   - In the `Data Subset` panel, select `Level` tab and select the `500 hectopascal` level then click the `Create Display` button.
   - Start the animation.
   - From the control window for this display, change the level to 250 hectopascals using `Levels` selector.
   - Rotate the display to see that the level you selected is displayed above the surface parameters.
   - Try other levels and notice how the pattern and position changes.
   - Stop the animation and return the display back to the first time in the sequence and reset to the overhead view.

3.2.4 Isosurfaces of Gridded Data

While plan views of 3D fields are useful, the three dimensionality of the atmosphere can be displayed in the IDV using isosurfaces. An isosurface represents a 3-D surface that connects points with the same value. On one side of the surface, values are greater than the surface value; on the other side, values are less than the surface value.

1. Create an isosurface of wind speed showing areas of winds 40 m/s or greater.
    - In the `Field Selector`'s `Data Sources` panel, select the `NCEP Model Data ETA data source` that you loaded at the start of this section.
    - In the `Fields` panel, expand the `3D grid` tab. Scroll down to the `Derived` tab to see a list of derived quantities. Select the `Speed (from GridRelative_u & GridRelative_v)` field.
    - In the `Displays` panel, expand the `3D Surface` tab and select `Isosurface`, then click the `Create Display` button.
    - Set the isosurface to display where winds of 40 m/s or greater exist. From the control window for this display, you can change the isosurface value either by typing a new value in the `Isosurface Value` entry box or by using the `Isosurface Value` slider widget () to the right of the entry box. Change the isosurface value to **40 m/s**.
    - Start the animation.
    - Rotate the display to see the wind speed pods from various angles.
    - Stop the animation and return the display back to the first time in the sequence and the view to the top.
    - Remove the display from the main window by clicking on the remove icon () or by selecting the control's **File Remove Display** menu.
2. You can also create isosurfaces of one variable that is colored by another variable. In this step, you will load in an isosurface of wind speed again, but this time you will have it colored by geopotential height.
    - In the `Fields` panel, expand the `3D grid` tab. Scroll down to the `Derived` tab to see a list of derived quantities. Select the `Speed (from GridRelative_u & GridRelative_v)` field.
    - In the `Displays` panel, expand the `3D Surface` tab and select `Isosurface colored by another parameter` then click the `Create Display` button. (you may have to scroll the `Displays` pane to find the display selection)
    - After you click the `Create Display` button, a dialog will pop up to allow you to select the `Other parameter`. From the dialog, expand the `3D grid` node and select the `geopotential height` field and click the `OK` button.
    - Adjust the wind speed value to **40 m/s**.
    - Start the animation.
    - Rotate the display to see the wind speed pods from various angles.



- ♦ Stop the animation and return the display back to the first time in the sequence and the view to the top.

3. You can modify the colors the display through the menus in the control window or legend.
   - ♦ You can use a different color table to show the variation of windspeed with altitude. Click on the `default` button next to the `Color table` label in the control window to bring up a list of pre-defined color tables. Select the **Basic** menu and choose the **Bright38** menu item to change to that color table. In a later exercise, you will learn how to create and modify your own color tables.
   - ♦ Rotate the display so you are looking at the view from the south. On the first time-step, the pod of 40 m/s winds bulges down into the blues of the color table. Place your cursor at the bottom of this bulge and note the Altitude readout at the bottom of the main window. It should read about 5000 m.
   - ♦ Now move your cursor over the color bar in the legend to the point where the color at the bottom of the bulge is. Notice that the readout on the color bar is about 5000 m as well.

   - ♦ Start the animation.
   - ♦ Rotate the display to see the wind speed pods from various angles.
   - ♦ Stop the animation and return the display back to the first time in the sequence.

3.2.5 Cross Sections of Gridded Data

Cross sections of data can be useful to show the variation of a field along a transect. In the IDV, a cross section display consists of a selector line and display in the main window, and a 2D display of the cross section in the control window.

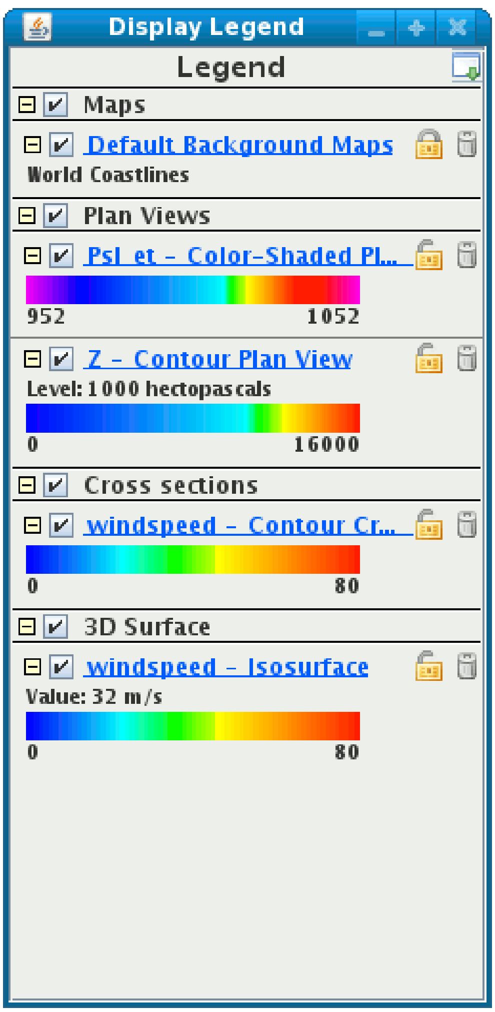
1. First, we'll create a vertical cross section of contours of wind speed and compare it to the isosurface made in the previous exercise.
   - Remove all displays except the isosurface display from the previous exercise.
   - In the `Field Selector`'s `Data Sources` panel, select the `ETA data source` that you loaded at the start of this section.
   - In the `Fields` panel, expand the `3D grid` tab. Scroll down to the `Derived` tab to see a list of derived quantities. Select the `Speed (from GridRelative_u & GridRelative_v)` field.
   - In the `Displays` panel, expand the Cross Sections tab and select `Contour Cross Section` then click the `Create Display` button. You could also create a color-filled contour or color shaded vertical cross section, but for the purposes of this exercise, we will use the contour cross section.
   - Move the cross section selector so it cuts perpendicularly across the wind speed core (N-S line through Iowa from Duluth, MN to New Orleans, LA). You can move the line by clicking and dragging the end selector point on the selector line.
   - Rotate the main display to the western view. Zoom in to get a better view of the contours. Note that the 40 m/s contour intersects the outer boundary of the isosurface.
   - Bring up the `Contour Properties Editor` from the `Settings` tab of the control window. Change the contour interval to 5, set the base to 30 and turn on dashing. Then click the `OK` button. The display changes so the contours are every 5 m/s and contour lines below the base (30) are dashed.
   - Start the animation. Note the southward progression of the jet core.
   - Stop the animation and return the display back to the first time in the sequence and the view to the top.
   - Remove the isosurface display from the main window. Do not remove the cross section display.
2. You can also create cross sections of 2D parameters that will show their variation along a transect. In this step, you will load in a plan view of sea level pressure and then a data transect of that field.
   - In the `Fields` panel, expand the `2D grid` tab. Select the `mean sea level pressure (ETA model reduction)` field.
   - In the `Displays` panel, select `Color-Shaded Plan View`. Then, hold the `Ctrl` key down and click on `Data Transect` to select both. Click the `Create Display` button.
   - You should see a color shaded display of sea level pressure and a new selector line for the data transect. Move the data transect selector so it spans from high pressure (reds) to low pressure (blues). Note how the display in the control window changes as you move it around.
   - You can share the position of cross section selectors between two different displays. In each of the control windows, (or by right clicking on the legend for each display) select the **Edit Sharing** menu and check the **Sharing On** checkbox. In the main view window, drag one end of the vertical cross section selector slightly. The selector for the data transect will snap to the same position as the vertical cross section line.
   - Start the animation. Is there any relation to the position of the jet core and the sea level pressure?

- Stop the animation and return the display back to the first time in the sequence and the view to the top.

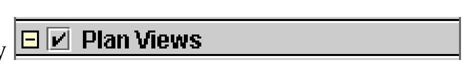
3.2.6 Using the Display Legends

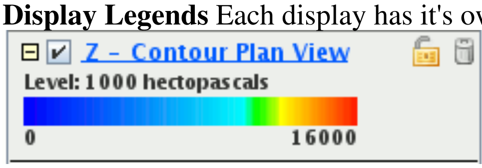
The Display Legend

When a display is created, an entry for it is added into the legend panel. There are two types of legends, side and bottom, that you can use to show the displays that are in a view. You can define which type of legend should be used. The side legend, shown below, is the default legend style.



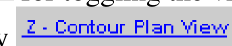
Main features of the side legend are:

- **Display Categories** Each display has a Display Category 

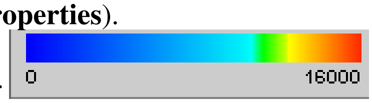
 that allows grouping of similar displays. The category can be changed as we saw in the previous exercise. Each category label includes a toggle button for collapsing ⊟ or expanding ⊞ the category details as well as a visibility checkbox ☑ for toggling the visibility of all displays in this category.
- **Display Legends** Each display has it's own legend


consisting of the following parts:

- A label for each display 

The label includes:
    - ◊ A toggle button for collapsing or expanding the display details
    - ◊ A visibility checkbox for toggling the visibility of this display,
    - ◊ The name of the display 

. Left clicking on this will bring up the the display's control window. Right clicking will bring up the control menu. The name of the display can be changed through labels through the `Display Control Properties` editor (**Edit Properties**).
    - ◊ A visibility toggle lock which locks/unlocks the toggling of visibility for this display.
    - ◊ A remove button for removing the display from the view window.
- Extra descriptive information about the display. 

This may include the full name of the parameter and ancillary information like the vertical level. You can add your own extra legend labels through the `Display Control Properties` editor (**Edit Properties**).
- Color bar 

(optional) - for colored displays, shows the color table being used and the range that the colors span. Right clicking on the color bar brings up a menu for changing the color table.

Things you can do with the side legend:

1. *Toggle visibility* - Uncheck the visibility toggle checkbox to make the display invisible. Click again to set it visible.
2. *Change the color table* - Right click on the legend color bar to bring up the color bar menu.
    - From the menu, select the **Solid Yellow** menu item.
3. *Show the Display Control Window* - Right click on the display label to bring up the display control menus. These menus are the same as those accessible through the control window. Select the **Control Window** menu item to bring the control window to the foreground.

3.2.7 Probing Gridded Data

The IDV has a variety of data probes that can be used with gridded data. Probes include value readouts, vertical profiles, model soundings, time/height cross sections and time series displays.

Let's create a data probe to read out various parameters at different levels.

- If you have not already done so, use the **Edit** **Remove All Displays** menu to remove the displays created in the previous exercise.
- In the `Fields` panel, expand the `2D grid` tab. Select the `mean sea level pressure (ETA model reduction)` field.
- In the `Displays` panel, select `Color-Shaded Plan View`. Expand the `Probes` tab, then, hold the `Ctrl` key down and click on `Data Probe/Time Series` to select both. Click the `Create Display` button. Once the displays are created, you can minimize the control window for the color shaded display.
- Drag the selector point around in the main window and notice how the readout changes.
- Now, let's look at the difference between *Nearest Neighbor* and *Weighted Average* sampling techniques.
    - Use the rubber band zooming feature (Shift+Mouse drag) on the main view window to zoom in over the central United States (centered on Kansas). The blocky pattern you see shows the outline of each grid cell.
    - Bring up the Control Window for the color shaded display and click the `Shade Colors` check box. Notice how the pattern changes. Return to the blocky pattern by unchecking the `Shade Colors` checkbox.
    - From the Probe Control window, right click on the `Psl_et` line and select the **Parameter: Psl_et** **Copy** menu.

  You will now have two entries for this field in the Probe control.
- Change one of the entries to sample using *Weighted Average* by clicking in the table row under the `Sampling` column and selecting that option from the dropdown list.
- Now move the probe around. Notice how the values in the two rows differ as you move from grid cell to grid cell in the display. *Nearest Neighbor* sampling will list out the value at the grid point. *Weighted Average* sampling will list out a weighted average of the values at surrounding grid points.

- Now, let's add in a 3D field to the list and sample that at different levels.

    - First, return the main display to its unzoomed position.
    - Right click on any row in the table and select the **Add** **parameter...** menu.
    - From the `Field Selector`, expand the `3D Grid` tab and select the `temperature` field. Click the `OK` button. You will now see an entry for **T** in the table in the Probe control window that lists the `Level` as *Probe's*. It will also be added to the chart.
    - Move the probe to the north and you will see that in general temperature decreases at the probe's altitude. Move it to the south and you will see that in general temperature increases.
    - Now rotate the display to a south view. Drag the probe up and down and notice how temperature changes with altitude in the atmosphere.
    - Now, let's move the temperature to it's own chart. Right click on the temperature line in the table and select the **Parameter T** **Chart Properties** menu select the same menu from the contrl's **View** **Parameters** menu.

- In the `Chart Name` box, type in **Temperature**. In the `Min` box, type in **-60** and for `Max`, type in **30** and click OK.
- Now add two more rows of Temperature to the table by copying the current temperature field.
- Change the `Level` of one of the temperature fields to be *850* hPa by clicking in the box for that row under the `Level` column and selecting that level from the list. Change the `Level` of another of the temperature fields to be *500* hPa.
- Now move the probe around and you will get a readout of each of the fields at the specified levels.
- Start the animation to see how the field values change with time.
- Stop the animation and remove all displays.

3.2.8 Probing Gridded Data (continued)

Now, let's look at other probes that can be used with 3D data - the Time/Height Display and the Grid Skew-T probe. The Time/Height display will show the variation of a field with height over time. The Grid Skew-T probe will display a model sounding generated from the data.

1. First, let's create a time/height display of relative humidity. We'll also load in an isosurface of this field for comparison. You can load in a probe without displaying the field in the main display.

   - If you have not already done so, use the **Edit Remove All Displays** menu to remove the displays created in the previous exercise.
   - First, we'll load in a time/height display of relative humidity. In the `Fields` panel, expand the `3D grid` tab. Select the `relative humidity` field.
   - In the `Displays` panel, select `Isosurface` (under the `3D Surface` tab). Then, hold the `Ctrl` key down and click on `Time/Height Display (Contours)` (under the `Probes` tab) to select both. Click the `Create Display` button. Once the displays are created, set the isosurface value to **80%** and minimize the isosurface control window.
   - Move the probe around to see how relative humidity varies with time and height at a particular location. Now, move the probe over Iowa.
   - Tilt the main display so you can see the structure of the relative humidity isosurface at the probe location, then step through the animation and compare the structure you see in the isosurface to the structure at the same time in the time/height display. How do they compare?
   - Stop the animation and return to the first time step.

2. Now, we'll create a Skew-T display of calculated model parameters.

   - In the `Fields` panel, expand the `3D grid` tab. Expand the `Derived` tab and select the `Sounding Data (with true winds)` field. This derived quantity will show up when the grid contains temperature, dewpoint and wind fields or any of these that can be derived from others.
   - In the `Displays` panel, select the `Grid Skew-T` display. Click the `Create Display` button.
   - Undock the `Time/Height` and the `Grid Skew-T` displays from the Dashboard. Arrange your windows so you can see the main display, the SkewT display and the Time/Height display.
   - Now, share the probe position with the Time/Height Display. In each of the control windows, use the **Edit Sharing Sharing On** menu to turn on sharing of the probe position. Move the probe over Iowa slightly to align the two probes.
   - Step through the animation and compare model temperature and dewpoint profiles with Time/Height display. In regions of high relative humidity, the temperature and dewpoint profiler will be close together.

3.2.9 Vector Displays

Vector components (e.g. winds, ocean currents, mantle plate velocities) can be displayed in the IDV in a variety of ways.

1. First, let's create a display of wind barbs with geopotential heights.
    - Clear out any existing displays.
    - In the `Field Selector`'s `Data Sources` panel, select the `NCEP Model Data ETA` data source.
    - In the `Fields` panel, expand the `3D grid` tab. Select the `geopotential height`, the `Contour Plan View` display and set the `Level` to be 500 hectopascals, then click the `Create Display` button.
    - In the `Fields` panel, expand the `3D grid` tab. Scroll down to the `Derived` tab and expand that. Select the `Flow Vectors (from GridRelative_u & GridRelative_v)` field.
    - In the `Displays` panel, select `Wind Barb Plan View`, in the `Level` tab, select 500 hectopascals, then click the `Create Display` button.
    - The display is pretty cluttered. You can change the skip interval to declutter the display. Try various intervals (found on the 'Display' tab of the dashboard).
    - Toggle between vectors and streamlines and play around with the streamline density slider.
    - Link the two displays by selecting the **Edit** **Sharing** **Sharing On** menu.
    - Change the level and see how the pattern changes.
    - Turn on the time animation.
2. Now, let's add a cross section display of wind vectors
    - Stop the time animation.
    - In the `Field Selector`'s `Fields` panel, select the `Flow Vectors (from GridRelative_u & GridRelative_v)` field.
    - In the `Displays` panel of the `Field Selector`, select `Vector Cross Section` and then click the `Create Display` button.
    - Move the cross section around and see how the flow changes.
    - Start the animation.

3.2.10 Working with Large Grids

Displaying high resolution grids in the IDV can be memory intensive. If you are reading from a remote server, you also have to deal with the latency of reading the grids across the internet. There are several strategies you can use to reduce the amount of memory used by the IDV.

1. First, let's load in a high resolution grid.
   - If you have not already done so, use the **Edit Remove All Displays and Data** menu to remove the displays created in the previous exercise.
   - In the `Catalog` data chooser, select the **Main IDV Catalog Unidata IDD Model Data UCAR(motherlode) North American Model NCEP NAM CONUS 12 km latest NCEP NAM CONUS 12km** dataset.
2. Now, let's subset the grid.
   - Open the `Data Source Properties` editor for `Latest NCEP NAM CONUS 12km`.
   - You can subset the grid by time and/or by coverage.
     - Click on the `Times` tab. Uncheck the `Use All` box. Right click on the panel and select the **Select every third one** menu option.
     - Click on the `Spatial Subset` tab. You can spatially subset the grid by stride (e.g. every nth point) and/or by region. Set the `X Stride` to `Every third point`. Notice that the `Y Stride` is linked to the `X Stride`.
     - Optionally, keep the stride at `All points` and select a sub-region on the map.
   - In the `Field Selector`, select the `3D grid->Temperature @ pressure` field and the `Contour Plan View` display.
   - Since we are displaying a single level, you can further reduce the grid by selecting a level instead of the entire grid. In the `Level` tab of the `Data Subset` panel, select the `50000 Pa` (500 millibar) level.
   - Click the `Create Display` button.
3. For more information on performance tuning, see the Performance Tuning section of the IDV User's Guide.

Footnotes:
Open the `Data Source Properties` editor by either:

- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

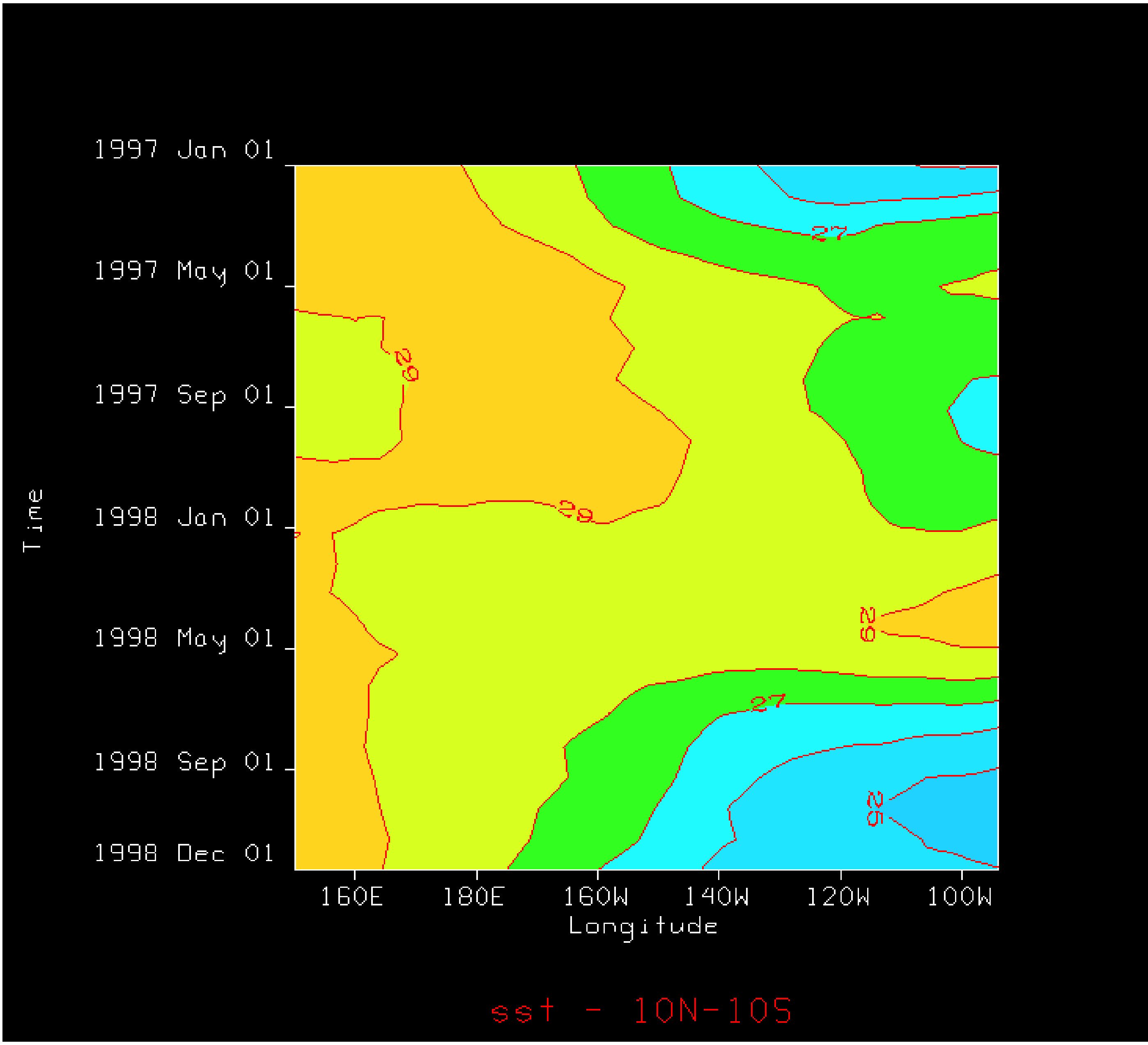
3.2.11 Hovmöller of Gridded Data

Hovmöller diagrams excel at displaying data as a function of time and location.

The Hovmöller diagram was invented by Danish meteorologist Ernest Aabo Hovmöller (1912-2008). Typically, time is displayed on the ordinate (y-axis) and geolocation on the abscissa (x-axis). They are commonly used in meteorology and oceanography.

Let's create a Hovmöller of sea surface temperature (SST) from the Pacific Ocean to observe El Niño/La Niña-Southern Oscillation.

- In the `Data Choosers`'s `URLs` tab, enter the `dods://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/noaa.ersst/sst.mnmean.` URL and click the `Add Source` button.
- In the `Field Selector`'s `Data Sources` panel, right-click the SST datasource and select `Properties` dialog.
- Click on the `Times` tab and select a time range from January 1997 to December 1998 and click the `Apply` button.
- Click on the `Spatial Subset` tab and select a region in the equatorial Pacific Ocean from Mexico to Hawaii and click the `OK` button.
- In the `Fields` panel, select the `Monthly Means of Sea Surface Temperature` field.
- In the `Displays` panel, expand the `Hovmoller` tab and select `Time-Longitude (Contours)` then click the `Create Display` button.
- Modify the contour labeling by clicking in the Contour `Change` button. Choose a contour interval of 1 degree Celsius.
- Modify the color table by clicking the Color table `default` button. Change the range from 15 to 32.
- Can you see the El Niño/La Niña-Southern Oscillation?


1997 Jan 01
1997 May 01
1997 Sep 01
1998 Jan 01
1998 May 01
1998 Sep 01
1998 Dec 01
Time
27
29
29
29
27
25
160E
180E
160W
140W
120W
100W
Longitude
sst - 10N-10S

3.2.12 Doing More with Grid Displays

1. Using the latest 80 km CONUS North American Model (NAM) data, create a display of color filled contours of sea level pressure, 500 hPa heights and isosurface of wind speed.
2. Display sea level pressure as contours draped over topography.
3. Create another color filled contour as topography display of sea level pressure and compare with the other one.
4. Use the **Edit Remove All Displays** menu to remove the displays.
5. Create display of geopotential height contours at 500 mb on a topography surface of the pressure from the latest GFS 80 km CONUS grid.
    - Create a color filled contour display of geopotential height over topography. When prompted for the topography field, select geopotential height.
    - Now do the same for contours over topography.
    - Turn sharing for both displays.
    - Set the level to 500 hPa.
    - Set the vertical scale to 4800 to 6000 meters to exaggerate the effect.
6. Create another color filled contour as topography display of geopotential height at 500 mb and compare with the other one.
7. Use the **Edit Remove All Displays and Data** menu to remove the displays and data.

3.2.13 Working with Ensemble Grids

Displaying ensemble grids in the IDV can be memory intensive. If you are reading from a remote server, you also have to deal with the latency of reading the grids across the internet. There are several strategies you can use to reduce the amount of memory used by the IDV.

1. First, let's load in a ensemble grid.
    - If you have not already done so, use the **Edit Remove All Displays** menu to remove the displays created in the previous exercise.
    - In the `Catalog` data chooser, select the **Main IDV Catalog Unidata IDD Model Data UCAR(motherlode) GlobalEnsemble Forecast System Model NCEP-GEFS-Global 1p0deg Ensemble with Ensemble Dimension latest NCEP-GEFS-Global 1p0deg Ensemble with Ensemble Dimension** dataset.
    - In the `Fields` panel, expand the `3D grid` tab and the `Mass` tab. Select the `geopotential height`, the `Contour Plan View` display.
    - In the `Fields` subset panel, and set the `Level` to be 500 hectopascals, on the `Times` tab, uncheck the `Use All` box, right click on the panel and select the **Select every third one** menu option , set the `Region` to conver `US CONUS`, on the `Ensemble` tab, select the `All Members`, or use the combination of Ctl and left mouse button and select a few members in the listand, then click the `Create Display` button.
    - User can change the range of color table to refelct how many members selected, the default range is 0 to 20, and the user can also disable "Color by Member" in the display control tab. The user will then have the capability of choosing an appropriate color table and color range for their data.
    - After creating the display, the user can specify irregular contour intervals with a semi-colon separated list of the values in the `Contour Property Editor`. For example: entering 5460;5800 would only show the 5460 and 5800 contour lines.
    - There are a few sample bundles in the Data/Grids/Ensemble folder.

3.2.14 Working with NcML

NcML is an XML representation of netCDF metadata, NcML is similar to the netCDF CDL (network Common data form Description Language), except, of course, it uses XML syntax..

1. A more advanced use of NcML is to modify existing NetCDF files. Displaying a bad netCDF grid dataset in the IDV can be frustrated. I am giving a example here that shows how to fix a bad nc file before loading it to the IDV.

```
netcdf X:/example/ecmf_2011032700_rh.nc {
 dimensions:
   x = 141;
   y = 61;
   z = UNLIMITED;   // (29 currently)
 variables:
   short ForecastHour(z=29);
     :units = "Degrees";
   float Longitude(x=141);
     :units = "Degrees";
   float Latitude(y=61);
     :units = "Hours";
   float RelativeHumidity_q25(z=29, y=61, x=141);
     :units = "Percent";
   float RelativeHumidity_q50(z=29, y=61, x=141);
     :units = "Percent";
   float RelativeHumidity_q75(z=29, y=61, x=141);
     :units = "Percent";

 :Title = "Relative Humidity Forecasts";
}
```

There are several issues in the above netCDF file, the unit of the variable Latitude and the the variable ForecastHour are switched, and dimension Z should be time dimension. The following NcML file will do correction of naming and unit.

```
<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2" location="./ecmf_20
  <dimension name="time" orgName="z" length="29" isUnlimited="true" />

  <variable name="Time" orgName="ForecastHour" shape="time" type="short">
    <attribute name="units" value="hours since 2011-3-27 00:00:00 UTC" />
  </variable>
  <variable name="Latitude" shape="y" type="float">
    <attribute name="units" value="Degrees" />
  </variable>
</netcdf>
```

NcMLExample.ncml

What has been modified here is the name of dimension z and variable ForecastHour. The unit of Latitude is also corrected, and the unit of new variable Time is `hours since 2011-3-27 00:00:00 UTC` which includes the information from the original file name. The above xml syntax is save in a file with an ncml file extension `ecmf_2011032700_rh.ncml`, and we can load this ncml file into the IDV for display. This is a very simple example, please visit NcML for complete information.

2. Another advanced use of NcML is to create virtual NetCDF datasets through aggregation. There are two `Data Source Type` in the IDV, `Aggregate Grids by Time` and `Aggregate WRF netCDF Grids by Time`, both using the built-in ncml to do the aggregation on the fly.

3.3 Displaying Satellite and Level III Radar Imagery

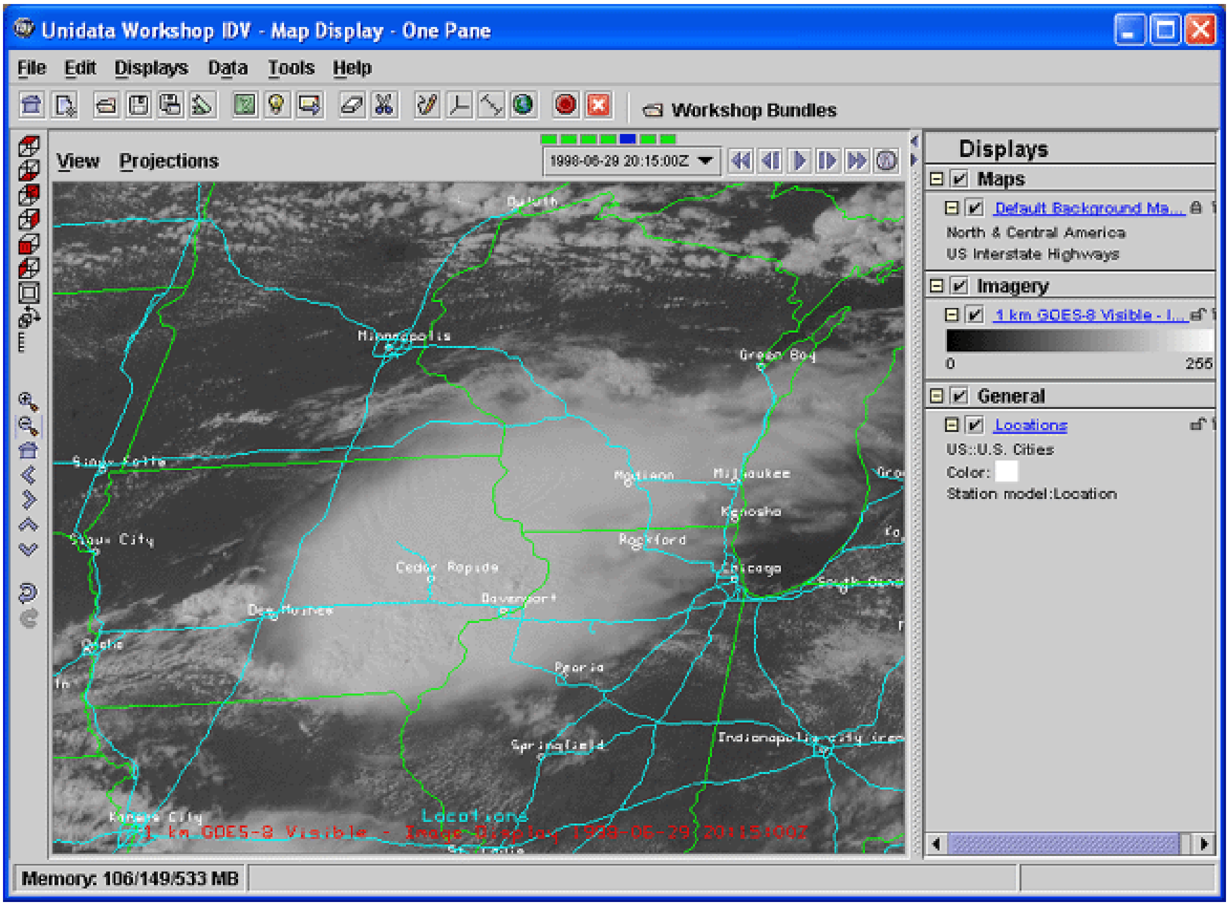
The IDV can load satellite imagery and Level III Radar Images from remote ADDE servers. Using ADDE allows the IDV to support multiple file formats. The IDV can also read in local McIDAS AREA formatted imagery.

3.3.0 Loading Satellite Imagery

Satellite imagery from geostationary and polar orbiting satellites is available through a network of servers supported by Unidata. Most of the imagery available on these servers covers the North America region. These servers use the Abstract Data Distribution Environment (ADDE) protocol.
First, we'll load in some visible imagery from the Bow Echo Case Study that we used in the **Gridded Data** section of the workshop.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Open the `Data Source Chooser`. Click on the `Images` tab.
3. We'll load in some visible satellite imagery. In the `Server:` selector, choose `adde.ucar.edu`. In the `Dataset:` selector, choose `CCS039`. If either of these is not in the list, just type them in. When you have made your selections, click the `Connect` button.
4. The IDV connects to the specified server and retrieves the types of images available in the dataset. From the `Image Type:` selector, choose the `1 km GOES-8 Visible` type.
5. You can either load in the last "*n*" images, or you can select by time. For this exercise, we want to select a subset of all available times, so click on the `Absolute times` radio button. When this option is selected, the server is queried for a set of available times for that type and these are listed in the `Times:` panel. Select the 7 images from `1998-06-29 16:15:00Z` through `1998-06-29 22:15:00Z`. Be sure that the `Create display` checkbox is selected and click the `Add Source` button.

6. Change the background maps using the `Default Background Maps Control`. Turn on the `North & Central America` and the `US Interstate Highways` maps. Make sure all other maps are off. When you are finished, go back to the `View Window`.
7. Use the **Displays Locations US U.S. Cities** menu to overlay some place names. Change the color of the location text to be white using the display control's **Edit Color** menu.
8. Start the animation. In this loop, you will see a large super cell thunderstorm develop over Iowa during this period. Stop the animation and show the 20:15Z image.
9. Zoom in over the super cell in eastern Iowa



10. This is visible imagery at 1 kilometer resolution. You can use the `Pixel Sampling` slider to see what the image would look like at a lower resolution.
    - ♦ Move the slider to **5** to see the difference an image at 6 km resolution would look like.
    - ♦ Return the slider back to **0** before proceeding to the next exercise.
    - ♦ Do not clear this display.

Footnotes:

Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

Open the `Default Background Maps` display control by either:

- Selecting the control's tab in the `Display Controls` tab in the `Dashboard`
- Left clicking on the control's label
- Right clicking on the label and selecting the **Control Window**

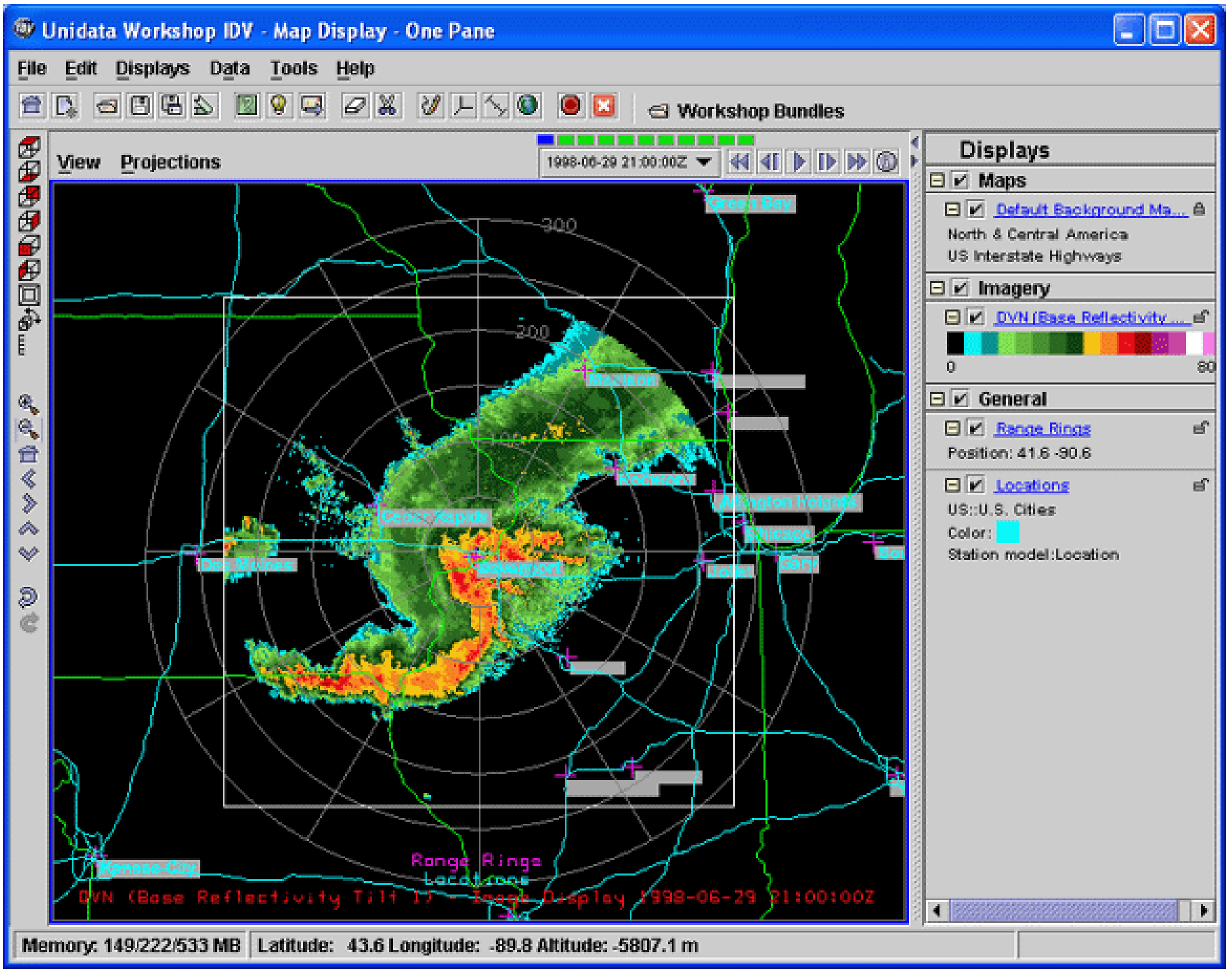
menu item.

3.3.1 Level III Radar Image Displays

Real-time Level III radar data is available from the network of ADDE servers in the Unidata community. At present, the IDV can read Level III data from both ADDE and TDS Radar servers.
In this exercise we'll display the Level III radar images from the ADDE server for the super cell area that we loaded in the previous exercise.

1. Open a new `View Window` by selecting the **File New View Window Map Display One Pane** menu. By default, the IDV will load data into the currently selected `View Window`. Minimize the other window with the satellite image in it for now.
2. Open the `Data Source Chooser.`. Click on the `Radar` under `Sat & Radar`, then the `ADDE Level III` under.
3. We'll load in Level III Data from the Bow Echo Case Study. In the `Server:` selector, choose `adde.ucar.edu`. In the `Dataset:` selector, choose `CCS039R`. If either of these is not in the list, just type them in. When you have made your selections, click the `Connect` button.
4. The IDV connects to the specified server and retrieves the types of radar data available in the dataset and the stations for which those data are available. On the station map, you will see **DVN** which is Davenport, Iowa. Click on this label to select it.
5. Using the `Radar Data Type:` selector, choose the `Base Reflectivity Tilt 1` type.
6. For this exercise, we want to select a subset of all available times, so click on the `Absolute times` radio button. With this option selected, the server is queried for a set of available times for that type and these are listed in the `Times:` panel. Select the 11 images from `1998-06-29 21:00:00Z` through `1998-06-29 21:59:00Z`. Be sure the `Data Type:` is set to `Reflectivity`. Make sure the `Create display` option is checked, and then click the `Add Source` button.

7. Change the background maps using the `Default Background Maps Control`. Turn on the `North & Central America` maps. the `US Interstate Highways` maps. Make sure all other maps are off. When you are finished, go back to the `View Window`.
8. Use the **Displays Locations US U.S. Cities** menu to overlay some place names. Change the display to show the locations using a custom layout by using the display control's `Layout Model` radio button in the control.
9. Overlay a radar range ring grid using the **Displays Special Range Rings** menu.



10. Start the animation. In this loop, you will see the bow echo move through the area. Restore the other view window from its minimized state. Notice that as the radar image loops through its steps, the corresponding satellite image is show in the other window. By default, animation steps are shared between views.
11. Stop the animation in the radar loop window. Now start the animation in the satellite image display. Notice this time, that the satellite display controls the images that are shown in the radar display.
12. Open the `Time Animation Properties` dialog in the satellite view window. Change the `Share Times` option to `None` and click the `OK` button.
13. Now start the animation in the satellite display and notice that the two views animate independently.
14. Stop the animation in both windows.
15. Close the window you were viewing the satellite data in by clicking the `X` in the upper right corner of the window.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

Open the `Default Background Maps` display control by either:

- Selecting the control's tab in the `Display Controls` tab in the `Dashboard`
- Left clicking on the control's label
- Right clicking on the label and selecting the **Control Window**

menu item.

Click the information icon in the `Time Animation Widget` toolbar.

3.3.2 Probing Level III Data

Just as we could probe gridded data, we can also probe the satellite and radar images.
First, let's add a data probe to the display we created in the previous exercise.

1. Complete the previous two exercises.
2. In the `Field Selector` window, make sure the `DVN (Base Reflectivity Tilt 1)` dataset is selected in the `Data Sources` panel and the `Image Sequence` field is selected in the `Fields` panel. In the `Displays` panel, select `Data Probe/Time Series`, and then click the `Create Display` button.
3. In the main view window, you will see a small red probe in the middle. Move the probe around over the echoes and notice the value changing in the `Data Probe` control window. Can't find probe?
4. Set the time animation to the first timestep.
5. Move the probe to the leading edge of the bow echo and turn on the time animation. Note that the value readout in the `Probe control` changes to show the value at that position for each time step.
6. Stop the animation and return to the first timestep.
7. Use the **Displays Special Range and Bearing** menu to add in the range and bearing tool. Position the left end point of the probe on the radar grid at 100km and 180 degrees.
8. Animate through the time sequence one step at a time, keeping track of where the cell that was under the left end point of the probe has moved to. Stop on the last timestep.
9. Position the right end of the range and bearing probe at the point where the cell has moved to. How fast did the storm move?
10. Remove all displays.

Footnotes:
If you can't see the probe, toggle off the radar image by clicking off the legend visibility checkbox for that display.

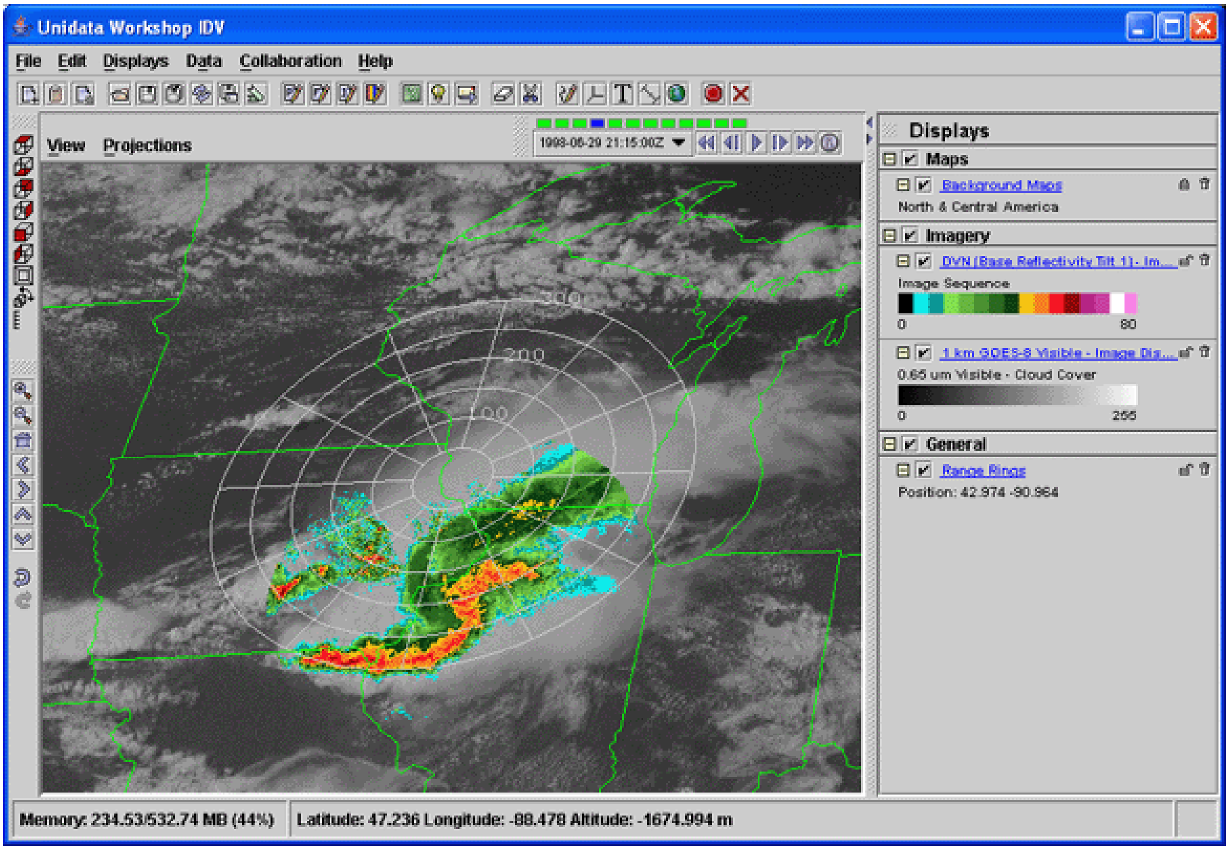
3.3.3 Overlaying Radar on Satellite and other imagery

You can create displays with both radar and satellite imagery in the same view as well as separate views. First, load up a couple of satellite images.

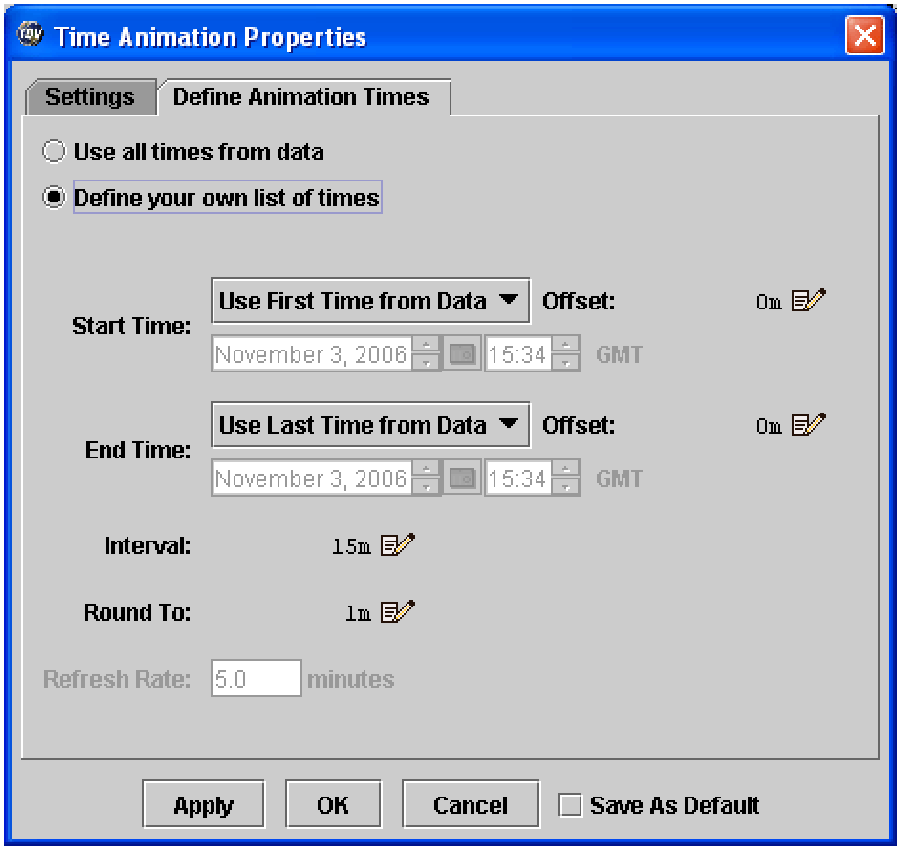
1. If you haven't already done so, remove all displays.
2. From the `Field Selector` window, select the `1 km GOES-8 Visible` data source.
3. Select the `0.65 um Visible - Cloud Cover` field in the `Fields` panel, and select `Image Sequence Display` in the `Displays` panel.
4. In the `Times` panel, uncheck the `Use Default` checkbox and select the 1998-06-29 21:15:00Z and 1998-06-29 22:15:00Z entries in the list.
5. Click the `Create Display` button to load in those two images.

Now, let's overlay some radar imagery.

1. First, let's keep the satellite projection for now. Uncheck the **Projections** **Auto-set Projection** menu item.
2. From the `Field Selector` window, select the `DVN (Base Reflectivity Tilt 1)` data source.
3. Select the `Image Sequence` field in the `Fields` panel, and select `Image Sequence display` in the `Displays` panel.
4. In the `Times` panel, make sure the `Use Default` checkbox is checked.
5. Click the `Create Display` button to load in the images.
6. Toggle between the two images by pressing the `F1` key.
7. Turn both images on by pressing the `F2` key.



8. You can reproject data into any projection you want. Let's change to the local radar projection.
   - Select the base reflectivity display control's **View Use Native Image Projection** menu item to switch to the radar projection.
   - Zoom in and out
9. You can also change the time animation steps to be more regular.
   - Open the `Time Animation Properties` dialog.
   - Select the `Define Animation Times` tab and then select the `Define your own list of times` option.



- By default, the time animation is automatically determined from the times of each dataset used in the display. You can change the default using this widget.
- For the `End Time` select the `Relative to Start Time` option.
- Click the edit icon () to the right of the `Offset:` label to set the offset from the start time. Set the offset to be 1 hour from the start time by typing **1** in the `hours` box, then click `OK`.
- Set the `Interval` to be **10** minutes and set the `Round to` option to **5** minutes.
- Click `Apply` and notice how this changes the time steps in the display.
- Try other combinations and see what effect they have.

10. Remove all displays.

You can also overlay radar and satellite images on top of a background image from a Web Map Server (WMS).

1. Use the **Displays Maps and Backgrounds Add Background Image** menu to load in the background image control. The default view is the Blue Marble image from NASA.
2. Change the layer type for this control to `Shaded Relief`.
3. From the `Field Selector` window, select the `DVN (Base Reflectivity Tilt 1)` data source.
4. Select the `Image Sequence` field in the `Fields` panel, and select `Image display` in the `Displays` panel. In the `Times` panel, make sure the `Use Default` checkbox is checked and then click the `Create Display` button to load in the images.
5. Toggle the images as you did above and use different layers in the control. We will look at this control in more detail later.
6. Remove all displays.

Footnotes:
Click the information icon in the `Time Animation Widget` toolbar.

3.3.4 Doing More with Image Displays

Try these exercises on your own.

1. Create a two panel map view window and run through the first two exercises in this section. Place the satellite imagery in one panel and the radar image in the other.
2. For the radar on satellite exercise, use the `US 2 km Composite Base Reflectivity` image data source from the case study.
3. Load in the latest ten 1 km GOES visible images from the `GINIEAST` ADDE image dataset. Center the display on Boulder (40N 105W) at a magnification of 1.

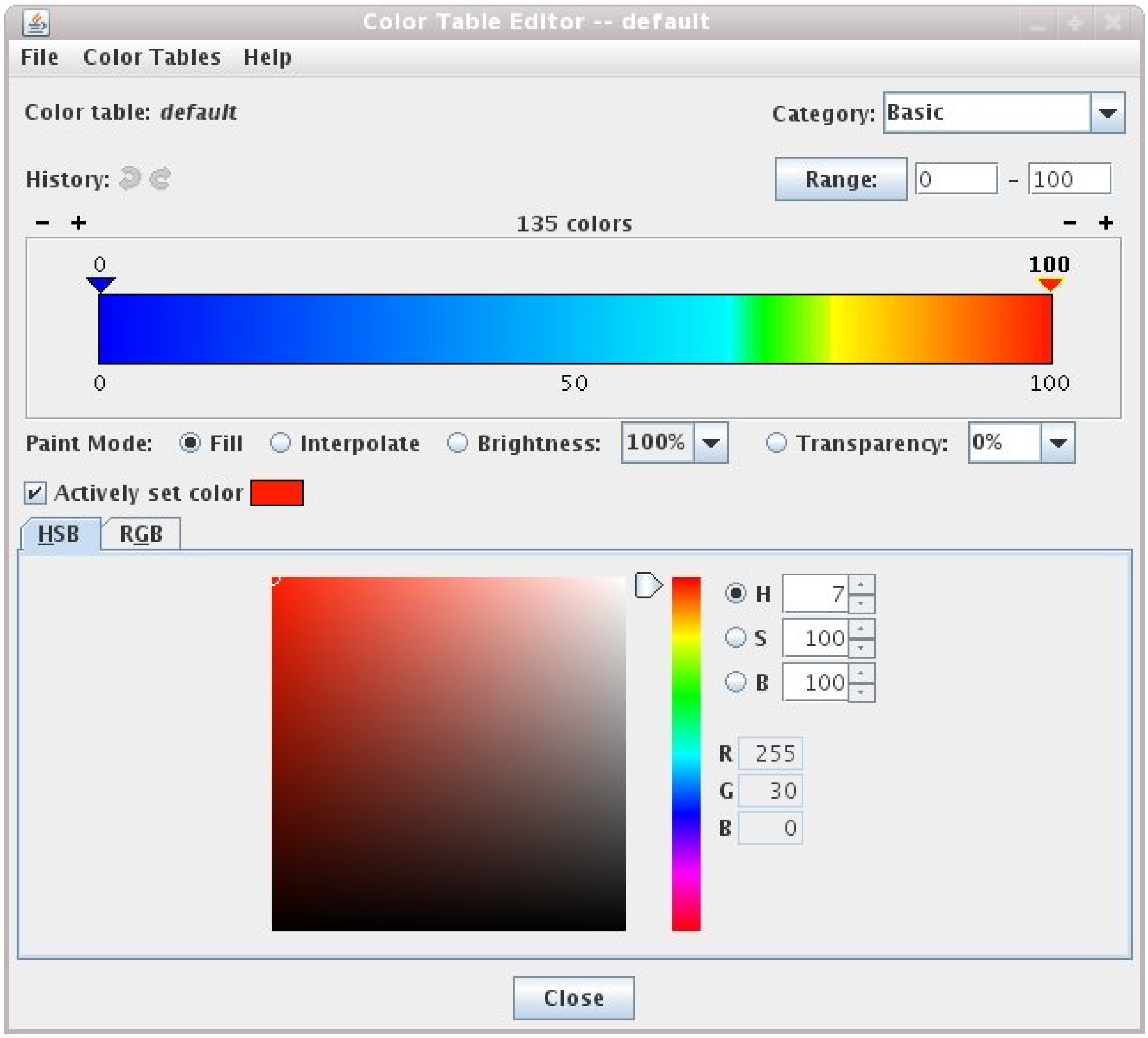
3.4 Color tables

In this section we will cover how to use the IDV Color Table Editor to view, edit and create color tables.

3.4.0 Introduction to the Color Table Editor

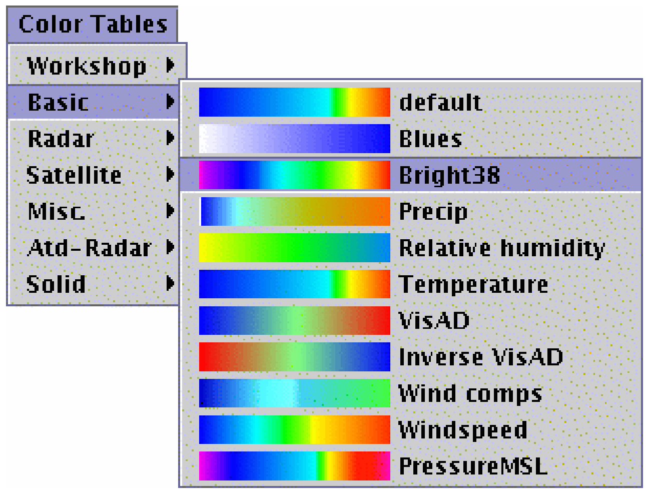
The color table editor allows you to view, modify and create color tables.

1. Open up the Color Table Editor from the **Tools** **Color Table Editor** menu.
2. You should see something like this (well, plus a Brightness field):



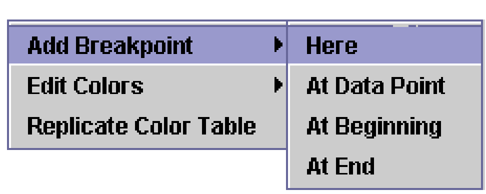
The Color Table Editor consists of:
- ♦ A set of command menus: **File**, **Color Tables** and **Help**.
- ♦ A color bar which shows the currently selected color table.
- ♦ A set of breakpoints that allow you to manipulate the color bar. (More about breakpoints)
- ♦ A color chooser which allows you to select a color.
- ♦ A set of paint mode items.
- ♦ History buttons that allow you to undo and redo changes.

3. First, we want to select a color table.
   - ♦ Select the **Color Tables** menu:

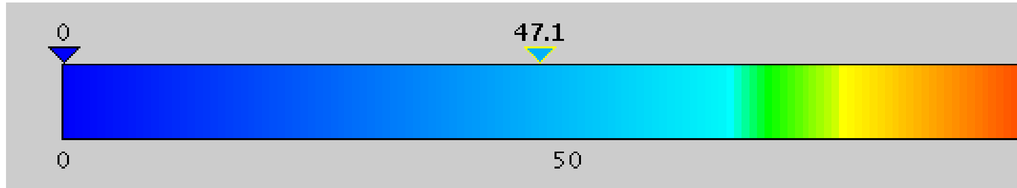


- Select **Basic** **Bright38**.

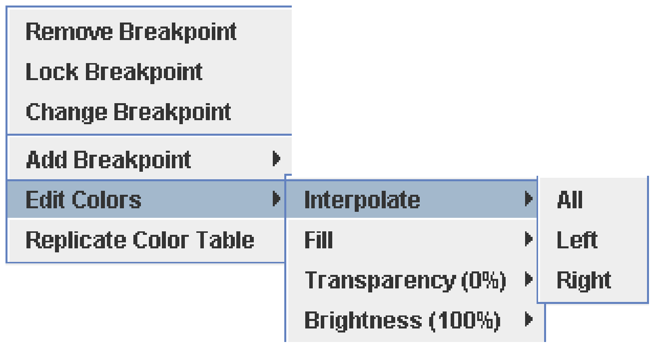
4. We don't want to change this particular color table so let's make a copy of it by saving the currently viewed color table under a new name.
   - Select the **File** **Save As...** menu item.
   - Enter the name for the new color table: "test1".
5. Now we'll play around a little changing the colors through direct manipulation. Notice when you mouse over the color bar the cursor changes to a paint brush. You "paint" with the selected color and paint mode.
   - Click the `Fill` paint mode button. You can select a color to fill with from the Color Chooser at the bottom of the window.
   - Click and drag in the color bar.
   - Try changing colors in the color chooser. Note - with the "Actively set color" button checked on the color below the selected breakpoint will be changed. Click on a breakpoint to select it and change the color.
   - Click the `Interpolate` paint mode button and paint a bit.
   - Click the `Brightness` paint mode button and change the brightness to 50%.
   - Click the `Transparency` paint mode button, select 50% transparency from `Transparency` menu.
   - Click on the `History` back and forward arrows to undo/redo any changes you make.
   - Click the + and − signs above the color bar to change the number of color slots in the color table.
6. Let's start fresh.
   - Select the **File** **Remove** menu item. This deletes the currently viewed color table.
   - You should be viewing the initial color table.
   - Select the **File** **Save As...** menu item and enter the name for a new color table: "test1".
   - Now we'll change the colors through the **Breakpoint** menu.
     - ◊ Add a new breakpoint.
       - · Right click in the middle of the color bar to bring up the **Breakpoint** menu:



       - · Select **Add Breakpoint** **Here**.
       - · You should have a color bar that looks something like:



◊ Right click on the new breakpoint and we will get a different menu:



◊ Select **Interpolate Right**.
◊ Select **Fill Left**.
◊ etc.

7. Let's get back to the beginning.
   - Select the **File Remove** menu item. This deletes the currently viewed color table.

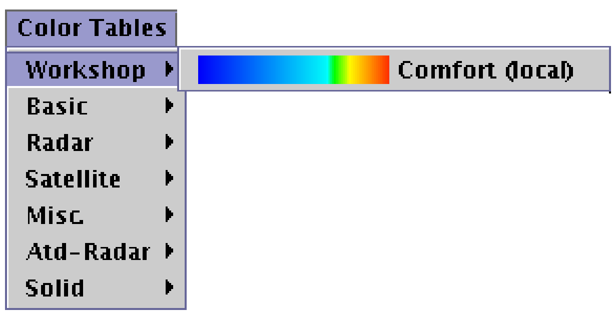
Footnotes:
Breakpoints are a color table editing convenience. You can add breakpoints to a color table, move them and remove them. You change the color in the color table directly by moving a breakpoint or by filling and/or interpolating between two breakpoints.

3.4.1 Creating a New Color Table

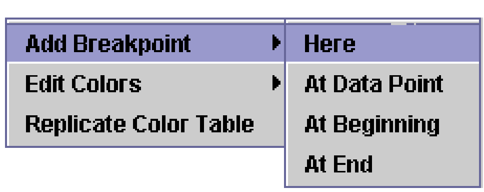
In this section we will create a color table to use within the IDV. This example color table is intended to represent a comfort range of temperatures.

1. Open up the Color Table Editor from the **Tools** **Color Table Editor** menu if it is not already open.
2. Select the *default* color table if it is not already displayed.
3. Rename this color table by selecting the **File** **Save As...** menu and entering the name "Comfort".
4. You should now have the same looking color table but with a new name displayed. We will create a new category, "Workshop" for this color table. More on categories
   - Click in the `Category` box and type "Workshop" and press return.
   - Now save this new color table by selecting the **File** **Save** menu item.
   - Verify that you indeed have a new category by selecting the **Color Tables** menu.

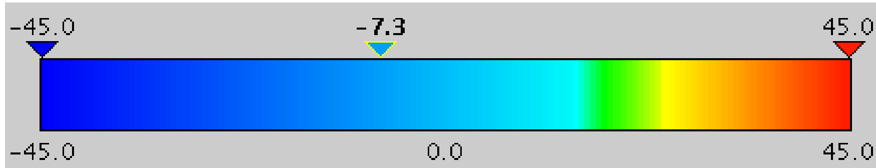
     You should see:

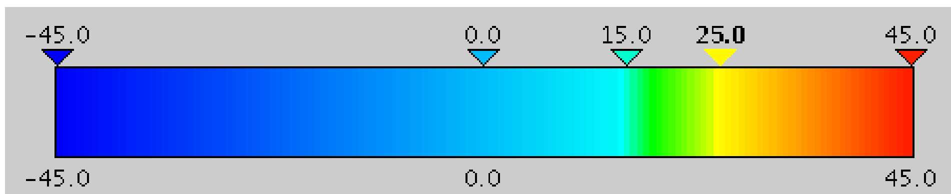
   Note: The label is "Comfort (local)" The "(local)" denotes that this color table is one that you have created, not a system provided color table.
5. Now, note the Range values *0.0 - 100.0*.
   The range values in the Color Table Editor are there for convenience purposes only. They allow you to see at what areas in the color table certain values would be shown in actual use.
   We are going to set these to an example range for temperature in Celsius.
   - Type in *-45* in the first box and press return.
   - Type in *45* in the second box and press return.
   - You should see the new range values displayed below the color box.
6. Add breakpoints at the 0 value (freezing), 15 (comfortable), and 25 (hot).
   - Right click on the color bar to bring up the **Color Bar Menu**:

   - Select **Add Breakpoint** **Here**
   - You should see a color bar with a new breakpoint in the middle:



- Click and drag the breakpoint to the 0 value. If you cannot get to exactly 0 then use either the right/left arrow keys or right mouse click on the breakpoint and select the **Change Breakpoint** menu item and enter the desired value.
- An alternative method would be to add a breakpoint at a particular value is to select the **Add Breakpoint At Data Point** menu item and enter the value 0.
- Repeat the above steps for the values 15 and 25.
- Your color bar should look like:



7. Now, let's come up with an interesting color scheme.
    - Click on the `Fill Edit Mode` button.
    - Select the center breakpoint (i.e., the "0" value point) by clicking on it.
    - Set the color under this break point to white by using the color chooser.
    - Right click on the center breakpoint and select the **Set Colors Interpolate Left** menu item.
    - How the rest of the Comfort color table is colored is an exercise left to the reader.
8. Once you are done you should save this color table by selecting the **File Save** menu item.

Footnotes:
Select the **Color Tables Basic default** menu item.

Color table categories have no actual semantic import. They are just a simple way to organize a (potentially) long list of color tables.

3.4.2 Using the New Color Table

Now we are going to use the new color table that we created in the previous section.

1. First, load in some data and an example display from the **Displays Favorite Bundles ToolBar Workshop Color table exercise** menu.
2. Note the color table that is used. By default this display uses the *Temperature* color table for the `temperature` field.
3. Change the color table for this display to the new color table, *Comfort* by clicking on the `Color Table` button to popup the Color Table menu.
4. Change the range.
    - Note the range that is being used is -90.0 to 45.0. This is the default range for `temperature` data. However, our *Comfort* color table was set up to use -45 to 45 as a range.
    - Change the range through the color table menu by clicking on the "Comfort" color table. You can also click on the legend's color bar or through the display control's **Edit Color Table Change Range...** menu. Enter the values -45, 45 and click the `OK` button.
5. You can quickly change the entire color table for this display.
    - Click on the "Comfort" color table to show the popup.
    - Select `Dimmer`.
    - Select `Brighter`.
    - You can also change transparency.
6. Bring up the `Color Table Editor` by clicking on the color bar in the control window.
    - Changes to this color table are automatically applied to the display unless the `Auto update` checkbox is unchecked.
    - Let's make our "comfort" zone be semi-transparent.
        - Select 80% transparency.
        - Right click on the breakpoint at 15 and select **Set Colors Transparency Right**.
        - Click the `Apply` button if `Auto update` is turned off.

3.5 WSR-88D Level II Data Displays

The IDV reads WSR-88D Level II data files. Each file has data for one complete volume scan of the atmosphere, including sweeps at several elevations. The IDV displays the data as sweeps and RHIs, in 2D and 3D, and as 3D isosurfaces. The IDV can be used for a general purpose Level II data viewer, and can also combine Level II data displays with other meteorological data, in 2D plots and in 3D displays in the upper atmosphere.

3.5.0 Loading WSR-88D Level II Radar Data

Level II data is accessed in the IDV from the remote THREDDS Data Server, each file having data from one WSR-88D radar for all sweeps (tilt) for one time.

Read more about Receiving and Storing Level II Data

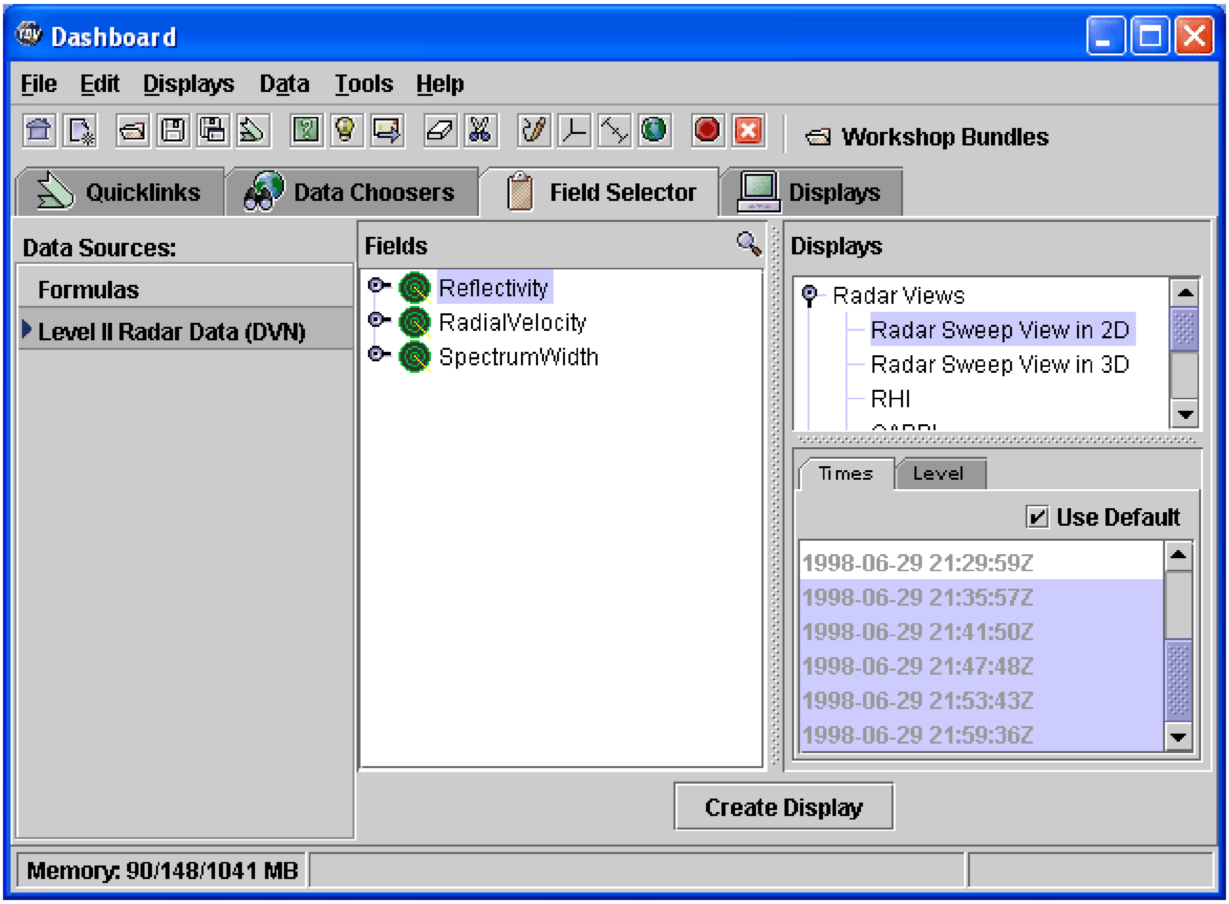
1. Setup for Level II displays
   - If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
   - Make sure the menu item **Projections Auto-set Projection** option is checked on.
   - Show the 3D wireframe box by toggling on the main menu item **View Show Wireframe Box**.
2. Open the Level II Data Source Chooser
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `Radar` tab. Then click on the `NEXRAD Remote` tab to see the Level II radar data remote chooser panel.
   - In the catalog selector drop-down, select the motherlode catalog that is included.

     http://motherlode.ucar.edu:8080/thredds/radarServer/catalog.xml

     If it is not in the list, type it into the catalog selector area and press enter. In the collections drop-down, select `Level II Radar for Case Study CCS039`

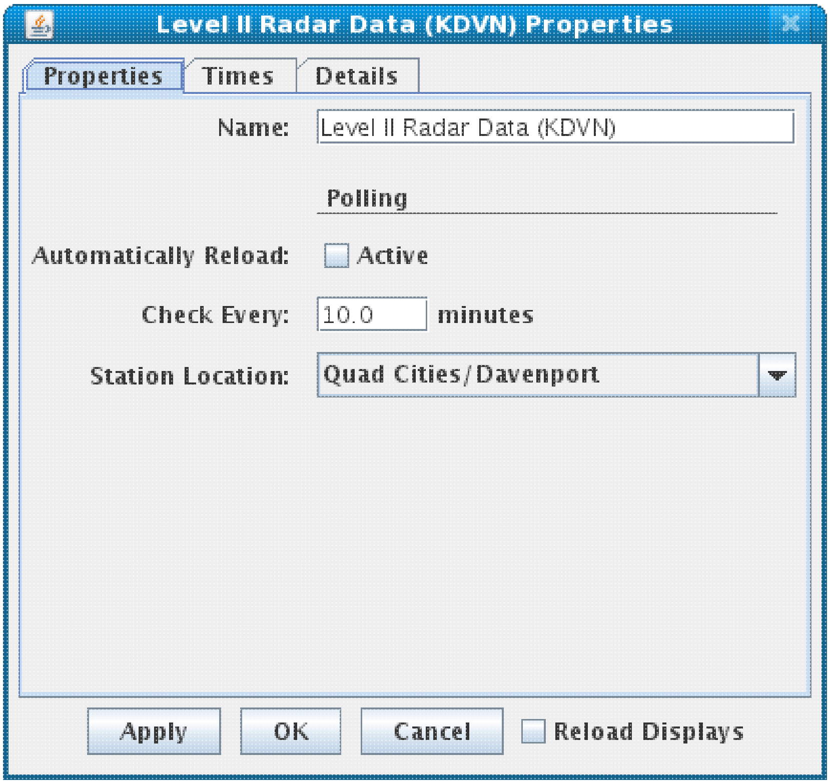
     ◊ Zoom in the map and select the station KDVN.
     ◊ Click on the `Absolute time` radio button. Select the images from `1998-06-29 21:00:00Z` through `1998-06-29 21:59:00Z`.
     ◊ Click the `Add Source` button.

The data source `Level II Radar (KDVN)` appears in the `Data Sources` list in the `Field Selector`.



Lets look at some of the properties of this dataset.

- Open the `Data Source Properties` editor for `Level II Radar Data(DVN)`.



- If you are looking at realtime data, you can set up the file polling information here to automatically reload new data as it becomes available.
- If the volume scans do not contain station information and you did not set the station information in the chooser, you can change the location using the widget in the properties dialog.
- Subset the times and choose the last five times from 1998-06-29 21:35:57Z to 1998-06-29 21:59:36Z and click `OK`.

Level II data has three `Fields` (radar data moments or data types). The IDV has several display types for Level II data. Any of the moments can be shown with any of the displays. For the following exercises, we will use Reflectivity.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

Open the `Data Source Properties` editor by either:

- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

Read more about Receiving and Storing Level II Data

The Level II data is supplied as volume-scan files, each file having all data from one WSR-88D radar for all sweeps for one "time". Unidata Community sites can receive Level II data using the Unidata Local Data Manager (LDM). Archived Level II data is available from the National Climatic Data Center (NCDC) (data from NCDC must be un-tarred and optionally, uncompressed for better performance).

If the files are stored on your file system with each station's files in a directory (folder) whose directory name is the station 4-character ID (e.g., KDVN for Davenport, Iowa). In many cases the data files do not have any location information in them and the IDV uses the directory name as a first guess at the station location. If you are working with data files where the directory name is not the station id, there is a way to specify the station in the IDV.

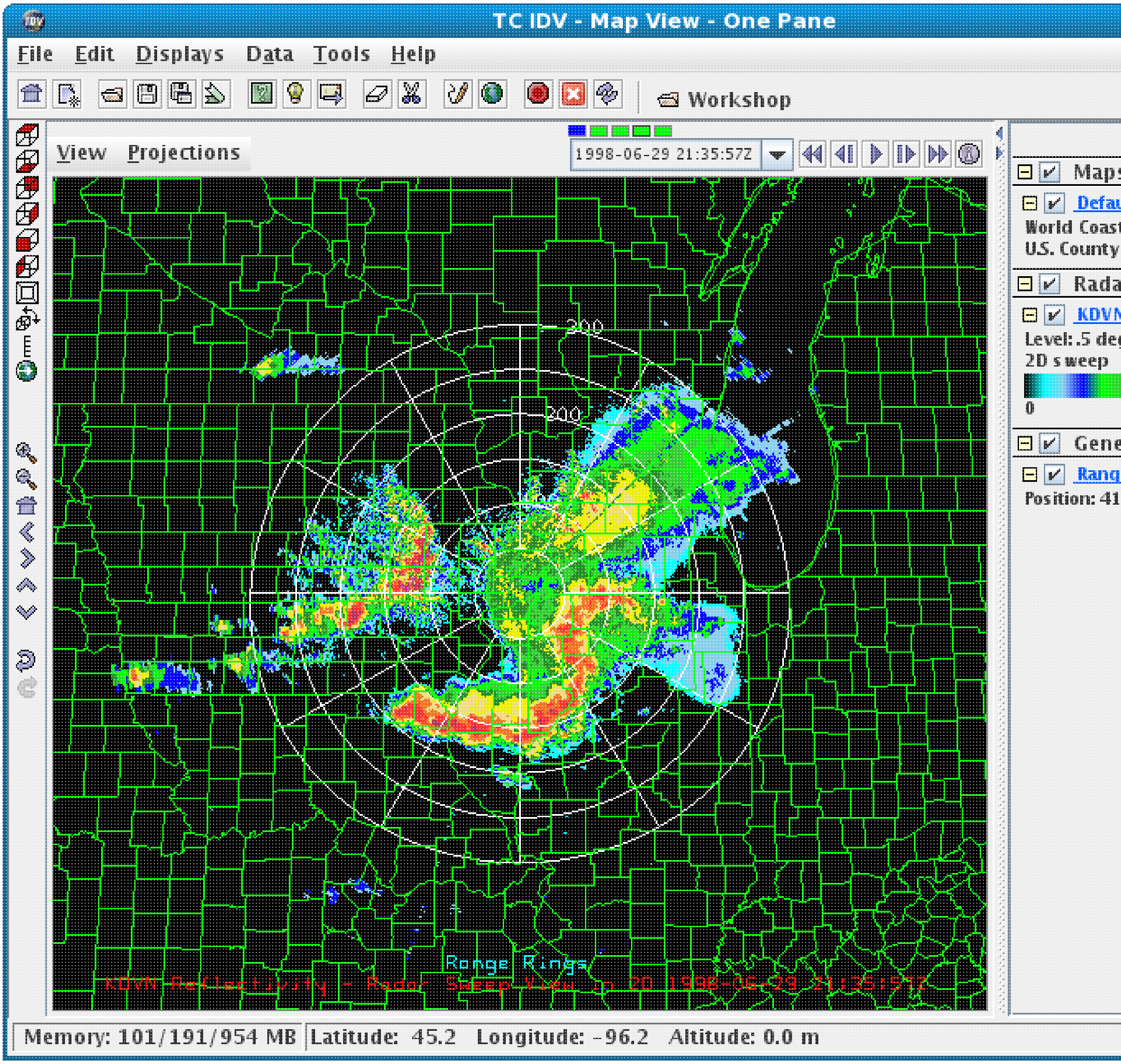
For the exercises in this section, we will use Level II data corresponding to the same times as the Level III data we used from the Bow Echo Case study in in the workshop dataset. See Installing local datasets for more information if you need to install these data.

3.5.1 Level II Sweep Displays in 2D and 3D

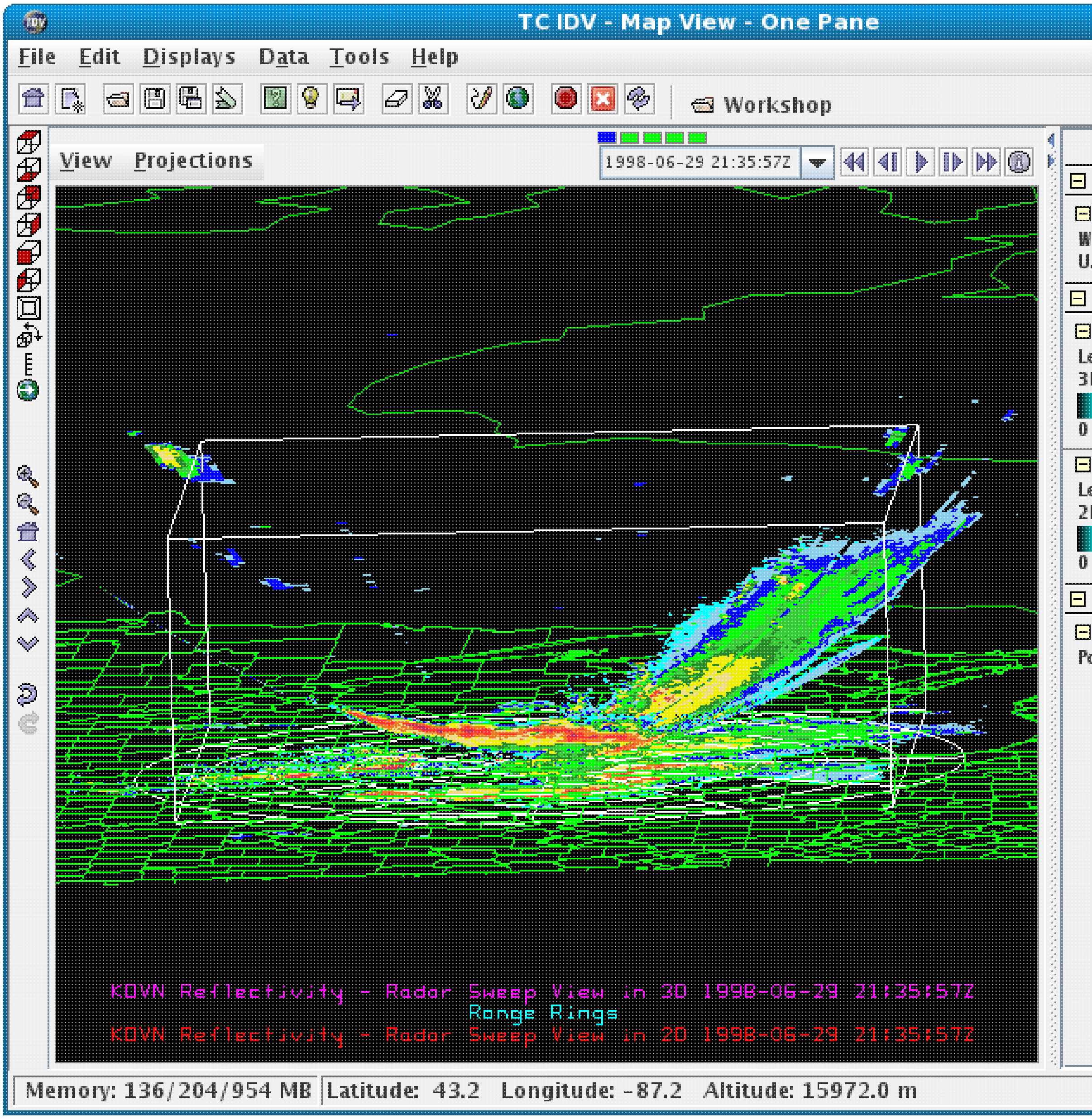
Sweep displays simulate conventional PPI radar plots of a single sweep of a map view, and also plot sweeps in the upper atmosphere,

If necessary, select Level II data and times as described in Accessing WSR-88D Level II Radar Data.

1. Select the `Reflectivity` field and create a `Radar Sweep View in 2D` display. A display of the Level II sweep at the lowest tilt appears in the main window.
2. Move to a new tilt angle in the volume scan; and put up counties map.
    - In the display's control window use the `Elevation Angles` selector to choose another tilt angle.
    - Zoom in to see the display character is that of the radar's native coordinate system, with wedge-shaped bins showing the radar range-azimuth coordinates.
    - Change the background maps using the `Default Background Maps Control`. Put on a map of U.S. Counties in green.
3. Add `Radar Range Rings`
    - Restore the initial display overhead view with the key combination Ctrl-r.
    - Select the **Displays Special Range Rings** menu from the `Main Menu Bar`. A set of range rings will appear, centered on the radar station.
    - Use the control widgets to change the spacing and color of the rings, the azimuth lines, visibility, and the labels.



4. Change the parameter (radar data moment) in the display.
   - In the control window `2D Sweep View`, select the **Edit->Change Parameter** menu item.
   - In the `Choose Parameter` panel, open the data source of interest by clicking on the tab.
   - Click on the parameter `Radial Velocity`, and click the `OK` button.
   - The display is recreated with the new parameter's data.
   - Switch the parameter back to `Reflectivity`.
5. 3D Sweep display
   - Add in a `Radar Sweep View in 3D` of `Reflectivity`.
   - The 3D display of the Level II sweep at the lowest tilt appears in the main window:



- ♦ Make sure both `Radar Sweep View in 2D` and the `Radar Sweep View in 3D` have the same `Sweep angle`.
- ♦ Rotate and zoom in as needed to see the character of the 3D display, and compare the two displays of the same data.
- ♦ If you have time switch the 3D sweep to a high tilt angle. The top of the wireframe box is at 16 km. Judge the maximum height of returned signals. About how far it is from the radar station?

6. Remove all displays.
7. Display the reflectivity as a Constant Altitude PPI (CAPPI). Select the `Reflectivity` field and create a `CAPPI` display.

   A CAPPI display of the reflectivity will appear in the main window.

Footnotes:

Open the `Default Background Maps` display control by either:

- Selecting the control's tab in the `Display Controls` tab in the `Dashboard`
- Left clicking on the control's label
- Right clicking on the label and selecting the **Control Window**

menu item.

3.5.2 Level II RHI Displays

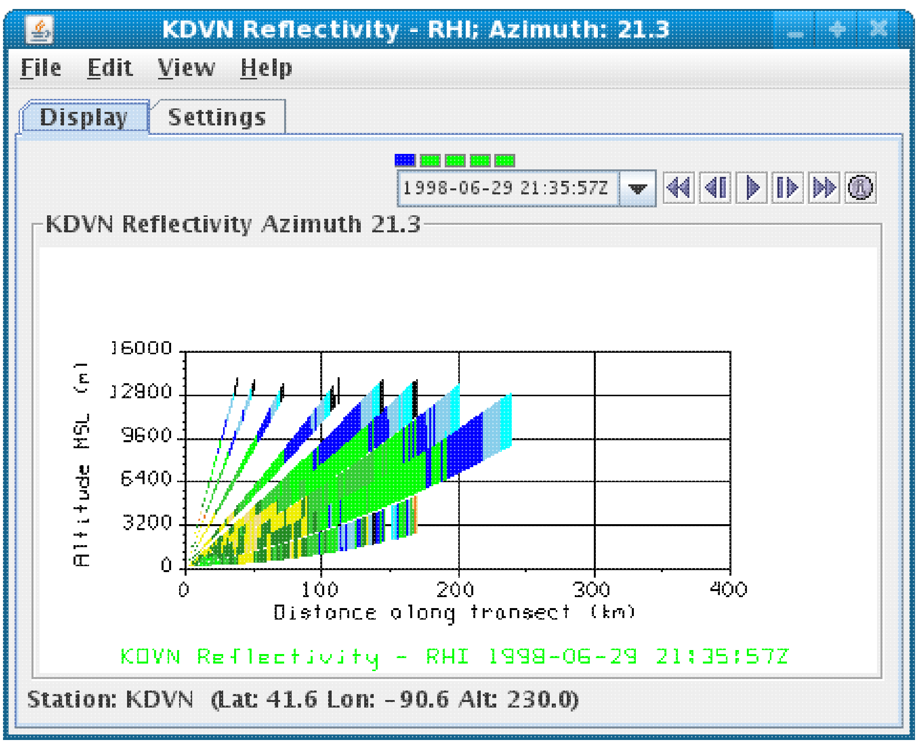
The IDV makes a pseudo-RHI display from Level II data. The display is presented in a conventional 2D plot, and in a 3D display at the correct position in the atmosphere.

The pseudo-RHI display is constructed from the several azimuthal sweeps available in the WSR-88D Level II output files.

If necessary, select Level II data and times as described in Accessing WSR-88D Level II Radar Data.

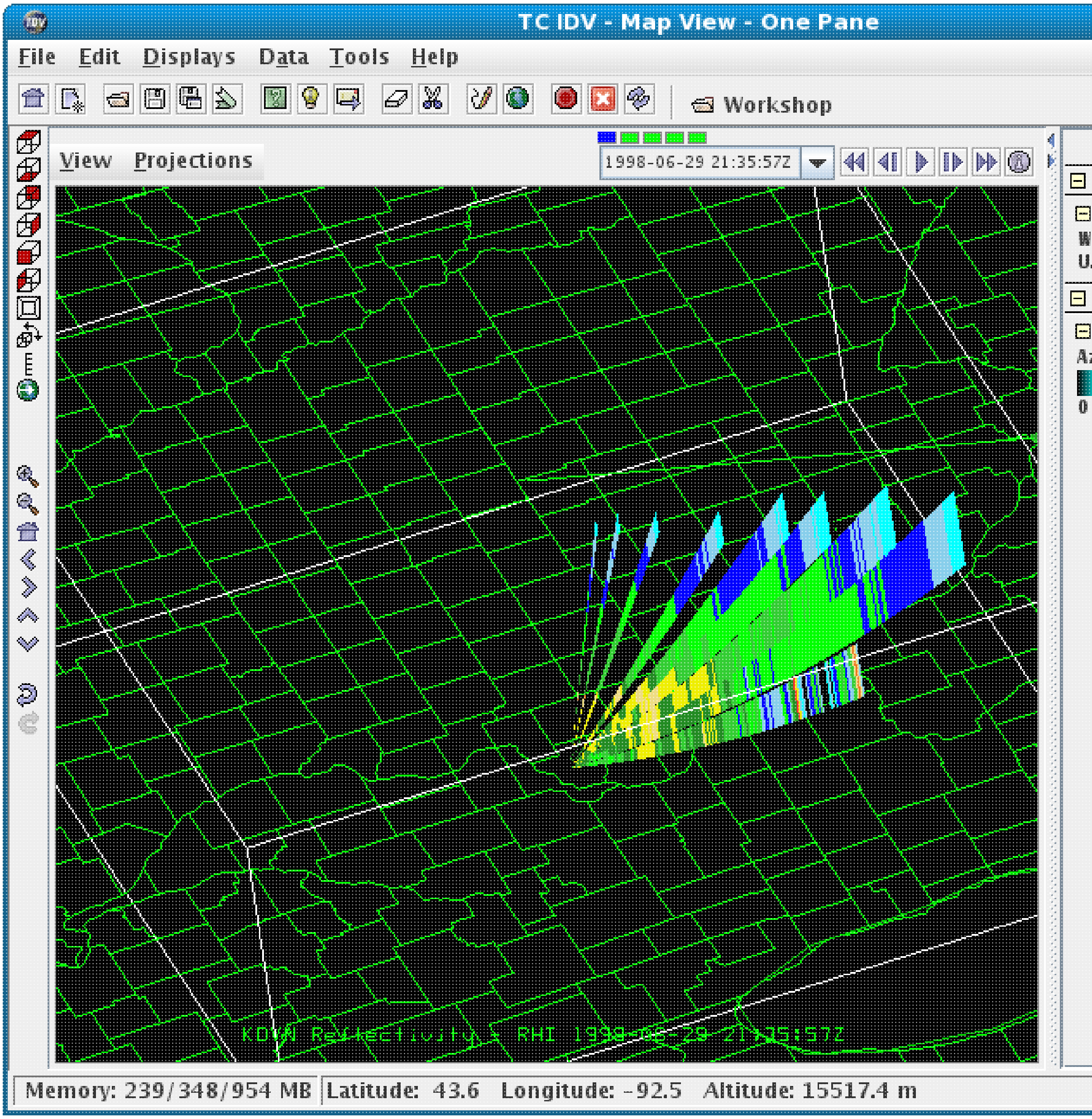
1. If other displays are in the main IDV window, select the **Edit Remove All Displays** menu item or the icon in the toolbar to clear them out.
2. In the `Field Selector` window create a `RHI` display using the `Reflectivity` field.

   The RHI control window appears in the `Dashboard`. The RHI display appears inside the RHI control window as a 2D plot.

3. The RHI also appears in the main 3D view window. View the RHI in 3D, and change the azimuth.

   - Rotate, pan, and zoom the 3D display as needed to see the RHI in 3D. Note the center and vertical width of the beams making up the "RHI" are in their actual locations in the atmosphere. You can show any Level II 3D radar display in the IDV with other upper air data, such as NOAA Profiler Network winds. The initial vertical exaggeration in the IDV is about 13:1.
   - Change the azimuth by dragging with the left mouse button on the small square box at the outer end of the selector line above the RHI.
   - If desired, restore the initial overhead viewpoint with the key combination Ctrl-r.



4. Working with the 2D RHI display
   - Pan the 2D display with a right mouse button drag, and zoom the display by dragging the right mouse button forward and backward with the Shift key pressed.
   - You can get the distance-height coordinate at any point on the 2D plot with the middle mouse button.
   - You can get the data value for a color by placing the mouse cursor anywhere on the color bar.
   - You can auto-rotate the RHI in both displays by clicking on the `Autorotate RHI` button. Click Autorotate on and watch both 2D and 3D displays.
   - Make a 3D sweep display for the same data, and watch the RHI move across the sweep.
   - Click off `Autorotate RHI`.

3.5.3 Level II CrossSection Displays

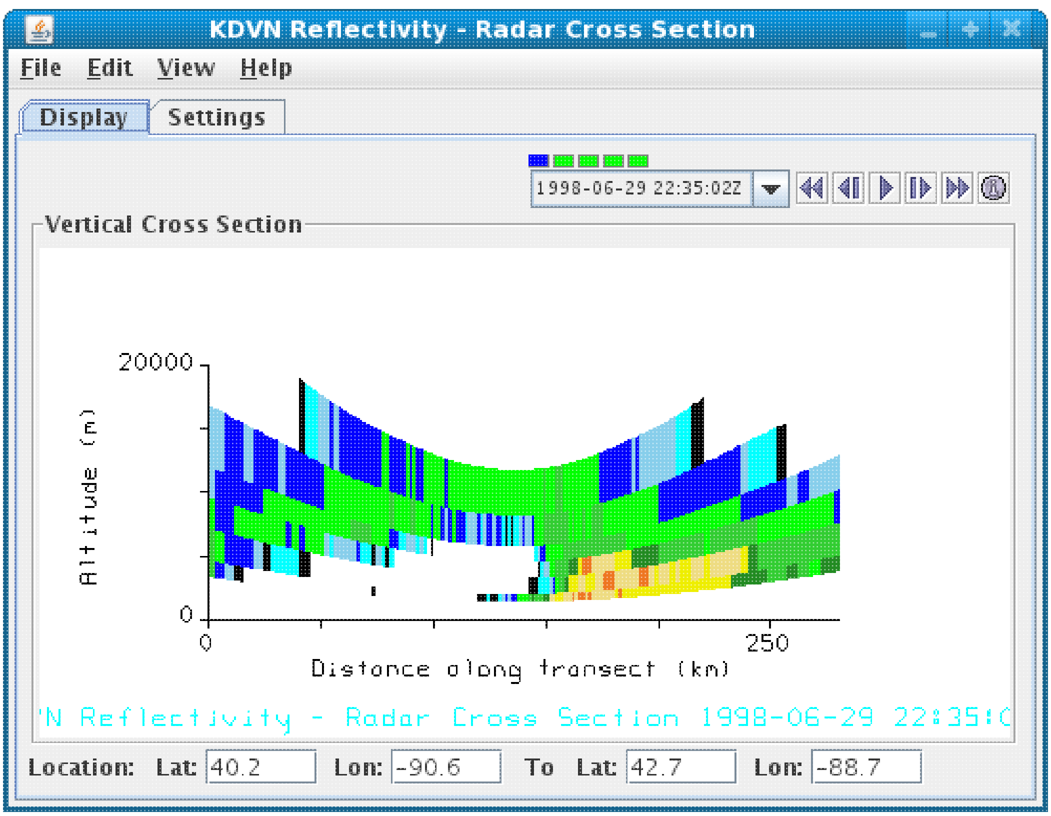
The IDV makes a vertical Cross Section display from Level II radar volume data. The display is presented in a conventional 2D plot, and in a 3D display at the correct position in the atmosphere.

The vertical Cross Section display is constructed from the cross section line and radials from several azimuthal sweeps available in the WSR-88D Level II output files.

If necessary, select Level II data and times as described in Accessing WSR-88D Level II Radar Data.

1. If other displays are in the main IDV window, select the **Edit** **Remove All Displays** menu item or the icon in the toolbar to clear them out.
2. Make sure the menu item **Projections** **Auto-set Projection** option is checked on.
3. In the `Field Selector` window create a `Radar Cross Section` display using the `Reflectivity` field.

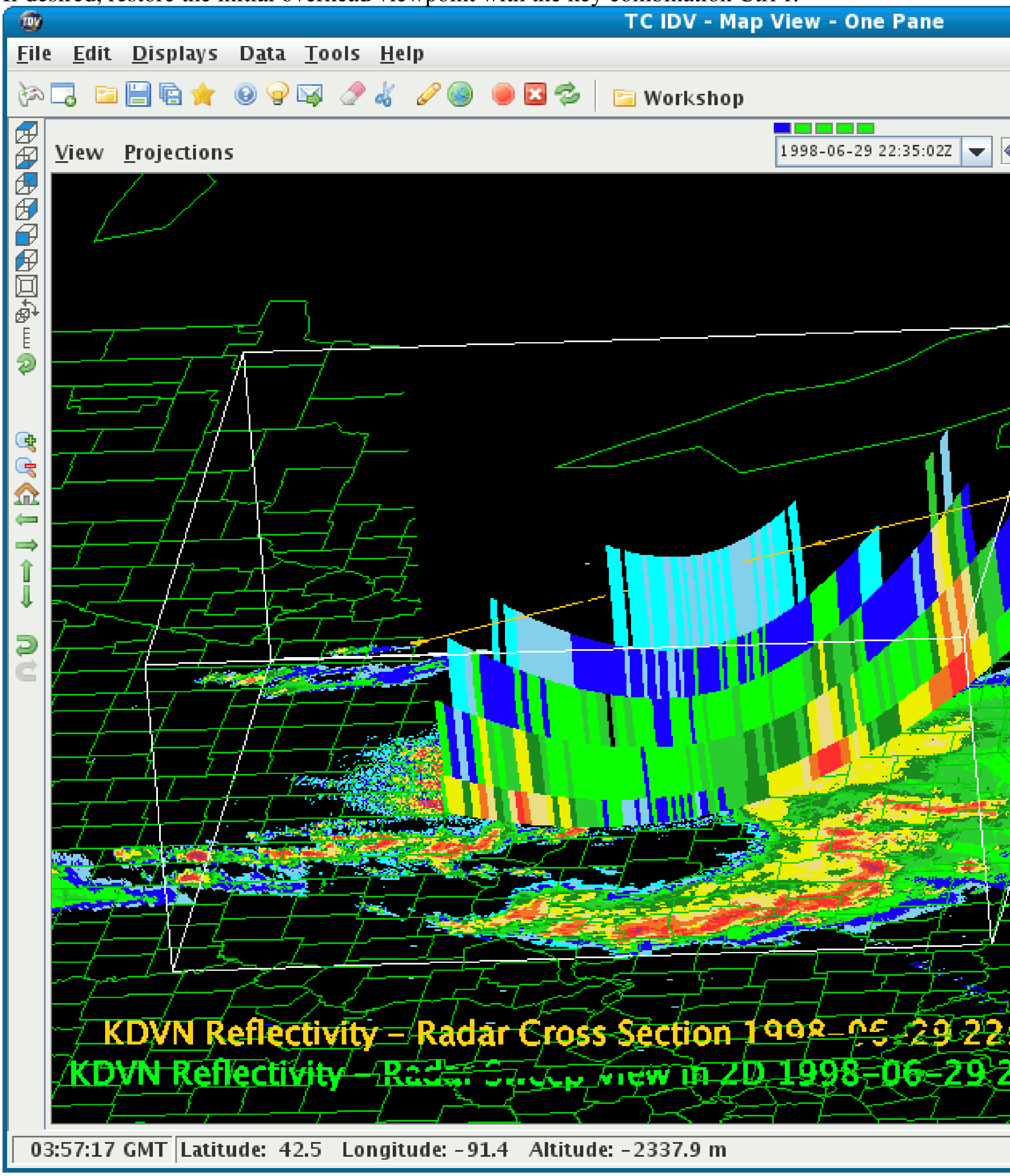
   The Cross Section control window appears in the `Dashboard`. The Cross Section display appears inside the Cross Section control window as a 2D plot.



4. The Cross Section also appears in the main 3D view window. View the Cross Section in 3D, and change the Cross Section line position.

   - Rotate, pan, and zoom the 3D display as needed to see the Cross Section in 3D. Note the Cross Section display is not a replacement of the RHI dislplay, if the number of intersections of the Cross Section line and radials of each sweep is low, the Cross Section display will have very few pixel points.

- Move the cross section selector so it cuts across the storm center You can move the line by clicking and dragging the end selector point on the selector line.
- If desired, restore the initial overhead viewpoint with the key combination Ctrl-r.



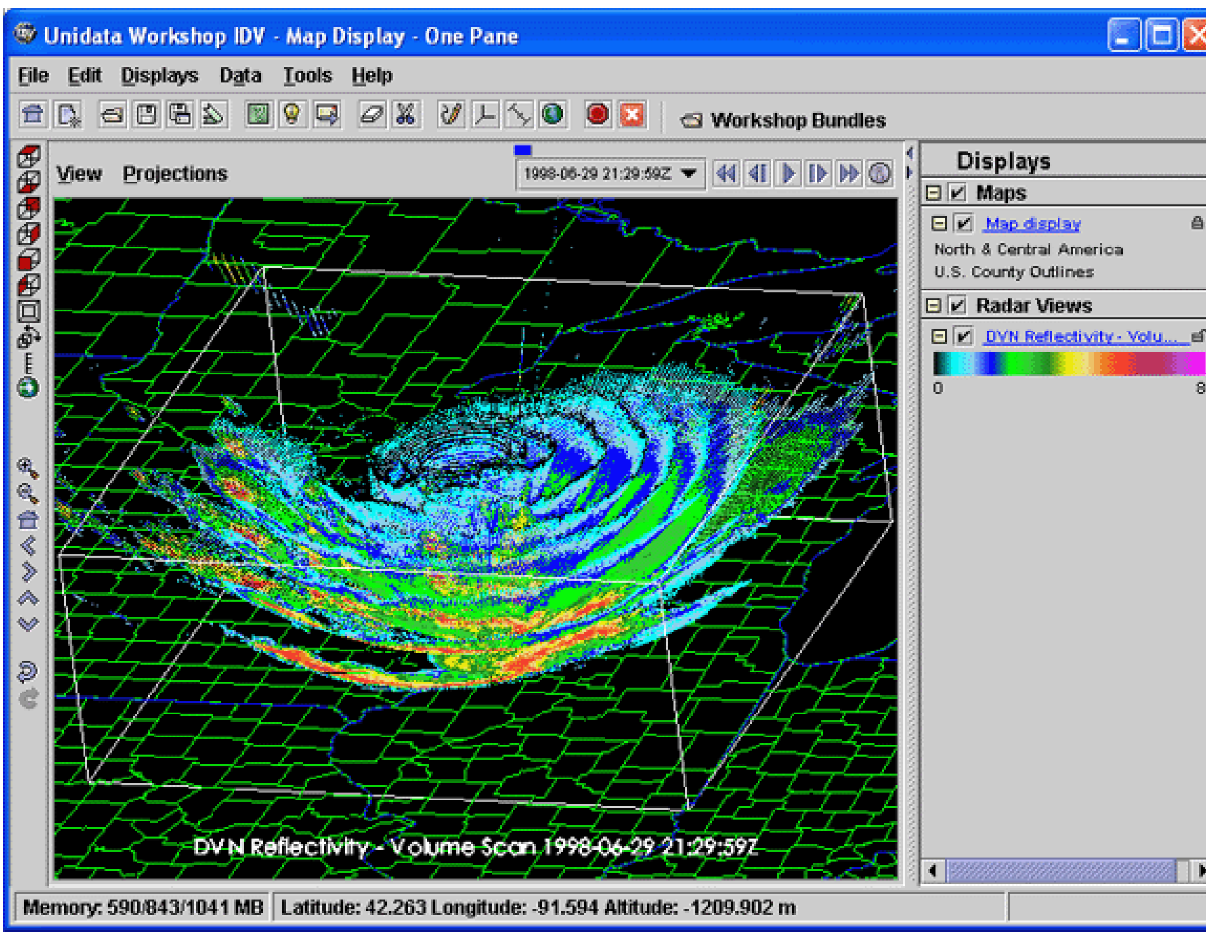
5. Working with the 2D Cross Section display
    - Pan the 2D display with a right mouse button drag, and zoom the display by dragging the right mouse button forward and backward with the Shift key pressed.
    - You can change the vertical scale of the 2D cross section view.
    - You can change the selector location by entering lat/lon of the end points of the cross section.

3.5.4 Level II Volume Scan Display

This IDV display shows all sweeps and individual bins in a WSR-88D volume scan, at their true positions in the atmosphere.

If necessary, select Level II data and times as described in Accessing WSR-88D Level II Radar Data.

1. If other displays are in the main IDV window, select the **Edit Remove All Displays** menu item or the icon in the toolbar to clear them out.
2. Select one of the times for this dataset.
   - Open the `Data Source Properties` editor for `Level II Radar Data (DVN).`
   - From the `Times` tab, select `1998-06-29 21:29:59Z` time.
   - Click the `OK` button.
3. In the `Field Selector` window create a `Volume Scan (all sweeps)` display using the `Reflectivity` field.
4. See the display in 3D.
   - The 3D display of all sweeps appears in the main window.



   In this display each bin value is represented by a single pixel colored by value. Since the sweeps are not solid images there is a degree of transparency.
   - Rotate to see the display in 3D. Zoom in to see individual data point values in single beams.

- ♦ Check the box next to the `Visible Range` label in the control and then click the `Change` button to change the range of data that is visible. Set the range to be from **30** to **50**.
- ♦ The resulting plot will show only the pixels where the values are between 30 and 50 dBz.

Footnotes:
Open the `Data Source Properties` editor by either:

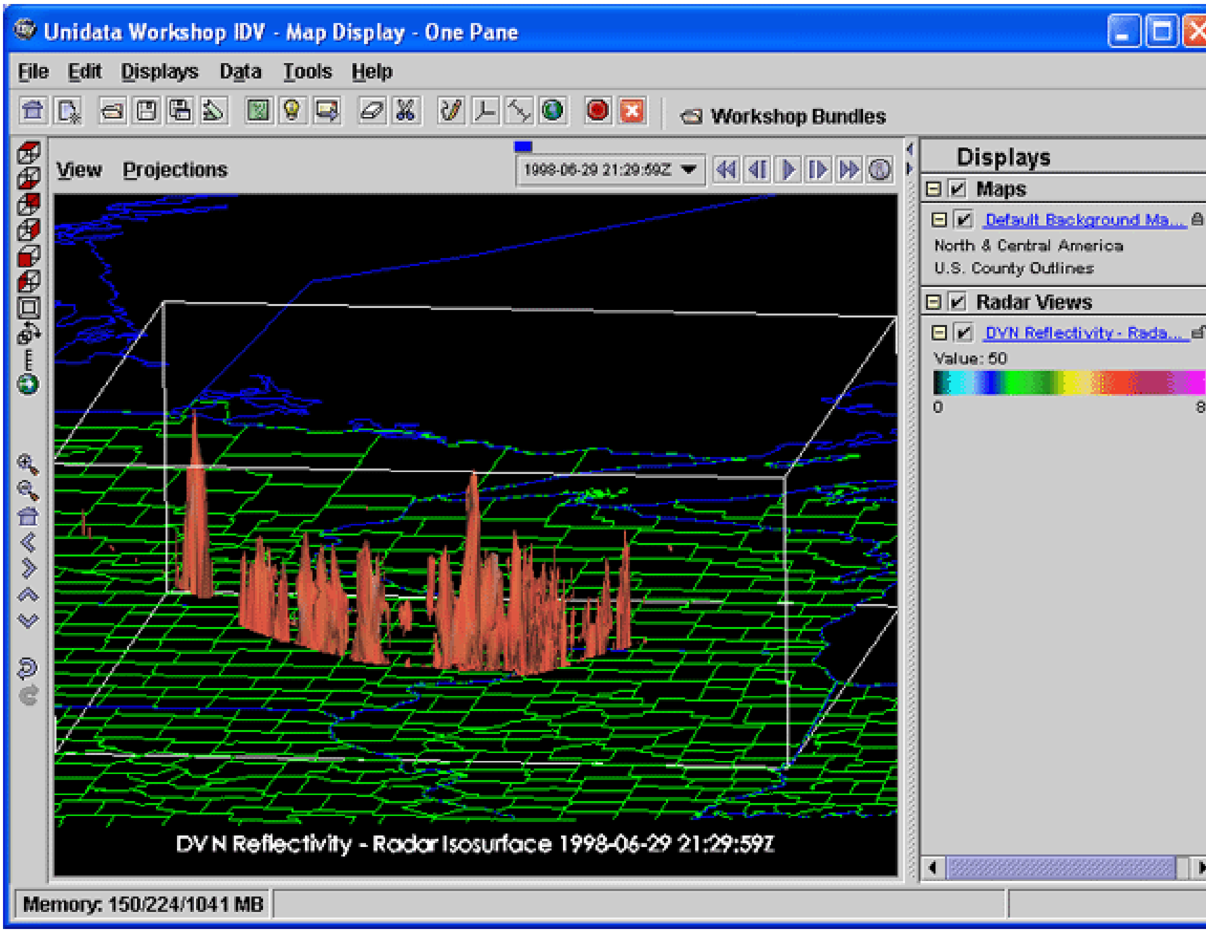
- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

3.5.5 Level II Isosurface Display

This IDV display shows isosurfaces of WSR-88D Level II volume scans.

If necessary, select Level II data and times as described in Accessing WSR-88D Level II Radar Data.

1. If other displays are in the main IDV window, select the **Edit** **Remove All Displays** menu item or the icon in the toolbar to clear them out.
2. If you have not already done so, select one time from the data source.
    - Open the `Data Source Properties` editor for `Level II Radar Data (DVN).`
    - From the `Times` tab, select `1998-06-29 21:29:59Z` time.
    - Click the `OK` button.
3. In the `Field Selector` window create a `Radar Isosurface` display using the `Reflectivity` field.
4. See the Display in 3D
    - A 3D display of the isosurfaces appears in the main window. It is complex because the initial isosurface value is low, and there are many low values in the display. What are Radar Reflectivity Isosurfaces?
    - Change the isosurface value to 50.0 dBZ with the `Isosurface Value` slider in the control window.
    - Rotate, pan, and zoom in as needed to see the new display in 3D.
    - What kind of weather system produced this radar result? What are the vertical shapes with isosurface value 50.0 dBZ? What does this arrangement of them mean? The IDV's default view has vertical exaggeration.



Footnotes:
Open the `Data Source Properties` editor by either:

- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

What are Radar Reflectivity Isosurfaces?

An isosurface is a 3D analog of a contour line. It shows the location of all data with a single data value. Interpolation is used between sweep altitudes in the IDV isosurface plot of Level II data. All data in a volume scan is used.

A radar isosurface from a line of thunderstorms, with surface value 50 dBZ, shows the location and shape of the 50 dBZ cores, to the precision permitted by several horizontal sweeps separated by a thousand or more meters in most cases.

3.5.6 Doing More with Nexrad Data Displays

Learn to do more with NEXRAD Data Displays

1. Access realtime NEXRAD data from the file system from the Radar Data Source Chooser
   - ♦ Open the `Data Source Chooser`.
   - ♦ In the `Data Source Chooser` click on the `Radar` tab. Then click on the `NEXRAD Local` tab to see the NEXARD radar data chooser panel.
   - ♦ From the `Relative Times` selector, select the `Use Most Recent 5 files`.
   - ♦ Use the directory (folder) selector to find the radar station data you want.
     - ◊ Go to the directory where the data are stored (e.g., `/data/gempak/nexrad/craft`).
     - ◊ Pick a station that has weather today.
     - ◊ Click the `Add Source` button.
   - ♦ Open the `Data Source Properties` editor for the dataset.
   - ♦ In the `Polling` section, check the `Active` option and set the `Check every` box to 5 minutes, then click `OK`.
   - ♦ Select the `Reflectivity` field and the `Radar Sweep View in 2D` display, then click `Create Display`.
   - ♦ Wait for 5 minutes and see that the display reloads with new data.
2. Access realtime and casestudies Level3 data from the remote server from the Radar Data Source Chooser
   - ♦ Open the `Data Source Chooser`.
   - ♦ In the `Data Source Chooser` click on the `Radar` tab. Then click on the `NEXRAD Remote` tab to see the NEXRAD radar data remote chooser panel.
   - ♦ In the catalog selector drop-down, select (or type in) the motherlode catalog this is included. In the collections drop-down, select `Level III Radar for Case Study CCS039`
     - ◊ Select a product from `product` drop down selector, or the first product will be selected
     - ◊ Zoom in the map and select the station KDVN.
     - ◊ Click on the `Absolute time` radio button. Select the images from `1998-06-29 21:00:00Z` through `1998-06-29 21:59:00Z`.
     - ◊ Click the `Add Source` button.
   - ♦ Display the casestudies Level 3 datasets.
   - ♦ In the collections drop-down, select `Level III Radar from IDD`
   - ♦ Display the real time Level 3 datasets.
3. Multiple Radars
   - ♦ Load Level II data for three or four times from three adjacent stations which have some detectable signals over the combined area.
   - ♦ Make 2D sweeps plots from all stations. Experiment to see if the order you display data from stations makes a difference.
   - ♦ Display the same data with 3D sweeps and rotate the view window to see intersecting sweeps.
4. Multiple Data Sources
   - ♦ Select a satellite data source that covers a region of the US that includes radars with echos.
   - ♦ Overlay the satellite image with Level II radar data.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

Open the `Data Source Properties` editor by either:

- Double click on the specific data source in the `Data Sources` panel in the `Field Selector`.
- Right click on the specific data source in the `Data Source` panel in the `Field Selector` and choose the **Properties** menu from the popup menu.

Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

3.6 Saving State, Views and Data

In this section we will explore how to save off the application state.

3.6.0 Creating and Using Bundles

In this section we will cover how to save the state of the IDV in a bundle file. Bundles are xml files that hold all of the application state of the IDV: data sources, displays, windows and view state.

1. First, clear all existing data and displays.
2. Load in a data source and create a display.
    - ♦ Load in the `Sample RUC Data` data source.
    - ♦ Create a `Color-Shaded Plan View` display with a `temperature` parameter.
    - ♦ Rotate the display in the main window and reposition the control window and the main window.
3. Save this state off as the `workshop.xidv` bundle file.
    - ♦ Select the **File** **Save As...** menu from the main menu bar.
    - ♦ Under the "What should be saved" leave everything checked.
    - ♦ Leave "No Jython" selected.
    - ♦ Make sure you are in the directory `/home/idv`. Enter the name `workshop.xidv`. Click `Save`.
4. You can use a bundle in a variety of ways:
    - ♦ Open a bundle while running.
        - ◊ Remove all displays and data. Reposition the main window.
        - ◊ Select the **File** **Open** menu item.
        - ◊ Select the `workshop.xidv` file.
        - ◊ You will be prompted whether you want to remove the current data and displays. Press `OK`.
        - ◊ You should see the state that was saved in the bundle.
    - ♦ You can also provide the bundle file as a command line argument when starting the IDV, e.g.:

```
./runIDV /home/idv/workshop.xidv
```

Footnotes:
To load in the `Sample RUC Data` data source:

- Open the `Catalog Chooser` by either:
    - ♦ Choose the **Data** **Choose Data** **From a Catalog** menu item.
    - ♦ Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Sample Data** **RUC Grid** item.
- Select the `Add Source` button.

To create a `Color-Shaded Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature` field.
- In the `Displays` list, select `Color-Shaded Plan View` and press the `Create Display` button.

3.6.1 More With Bundles

Let's do more with bundles.

1. Let's only save loaded data sources
   1. If you don't have any data and displays, load the `workshop.xidv` bundle with the **File** **Open** menu.
   2. Let's also load the Sample AVN Global grid data from the Catalogs tab.
   3. Select the **File** **Save As** menu item.
   4. Enter `justdata.xidv` as a file name.
   5. Uncheck `Views` and `Displays` and click `Save`.
   6. Open this bundle with the menu **File** **Open** and check remove all displays and data.
2. Let's only save the displays.
   1. From the field selector, create a Color-Shaded Plan View from the `temperature` field from the AVN Data Source.
   2. Save the bundle with the menu **File** **Save As**.
   3. Enter `justdisplays.xidv` as a file.
   4. Uncheck `Views` and `Data Sources`.
   5. Check `Displays`.
   6. Click `Save`.
   7. Remove the displays with the **Edit** **Remove all displays** menu item.
   8. Open the `justdisplays.xidv` bundle with the menu **File** **Open**.
   9. Make sure you **UNCHECK** "`Remove displays and data.`"
   10. When we load in a bundle with only displays the IDV will prompt you for the data.
   11. Choose the `temperature` field from the AVN data source.

3.6.2 The Default Bundle

You can create a default IDV state and save it using the bundle facility.

1. By default the IDV imports a `Default` bundle when it starts up.
2. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
3. Open up a new view window with the **File New View Window** menu item.
4. Change its state.
    - Change its perspective.
    - Select a different projection with the **Projections Predefined** menu.
5. Save the current state as the default bundle by selecting the **File Default Bundle Save** menu item from the main menu bar.
6. This file is saved in the user's local resource directory:
   `<home directory>/.unidata/idv/DefaultIdv`
   (e.g., /home/idv/.unidata/idv/DefaultIdv) as `default.xidv`.
7. Exit and restart the IDV. You should see the previously saved state.
8. Remove all data and displays.
9. Select the **File Default Bundle Open** menu item or the button from the toolbar to reload the bundle.
10. Delete the default bundle with the **File Default Bundle Remove** menu item from the main menu bar.
11. Exit and restart the IDV. You should see the previously saved state defined in your user preferences.

3.6.3 Using Favorite Bundles

In this section we will cover how to save the state of the IDV as a favorite bundle. A favorite bundle is a normal IDV bundle that is saved in a location that the application manages.

1. First, if you don't have any displays or data then load in a data source and create a display.
2. Save this state as a favorite.
    - Select the **File** **Save As Favorite...** menu from the main menu bar.
    - Enter: "Toolbar>Workshop" in the category field. The ">" defines sub-categories.
    - Enter: "Test" for the name.
    - Press `OK`.
3. View this favorite.
    - Remove all displays and data.
    - Load in the favorite by selecting the **Displays** **Favorite Bundles** **Toolbar** **Workshop** **Test** menu item.
4. Edit the favorites
    - Create 3 or 4 other favorites: e.g., Category="General" and Name="Some name 1", "Some name 2", etc.
    - Open the favorites editor with **Displays** **Favorite Bundles** **Manage**.
    - Right click on the favorite to show a popup menu.
    - Select **Delete**.
    - Drag and drop them to change their categories.
5. Take a closer look at some of the default system favorites
    - Open the **Displays** **Favorite Bundles** **Ocean** **El Nino/La Nina Comparison**
    - This shows temperature cross sections of the Pacific Ocean for normal, El Nino and La Nina years.
    - Move the cross section in the main view. See how each of the three cross sections are linked.
    - Note: This bundle changed the vertical scale of the main window. To continue with the exercises, either create a new window (using: **File** **New** **View Window**), reset the vertical scale (using **View** **Viewpoint** **Vertical Scale**), or exit and restart the IDV.

3.6.4 Data in Bundles

In this section we will cover how to save data inside a bundle.

1. Save a local file into a bundle
    - From the **Quicklinks History** page select the RUC data source.
    - Create some display with one of the fields.
    - Save this as a bundle with the **File Save As** menu entry. Instead of specifying a `.xidv` file suffix specify a `.zidv` suffix.
    - When asked which local data sources to include click on the checkbox.
    - In the gridded data dialog select the fields you want to save.
    - Load this zidv bundle in again. When prompted where to write the data files just press `OK`.
2. Save remote data locally.
    - Load in the `Davenport Radar` favorite from the `Workshop` menu.
    - View the `Details` tab in the Properties dialog for the `DVN` data source by right-clicking on the entry in the Field Selector. Note the ADDE URLs.
    - Right-click on the data source again and choose **Make Data Source Local**
    - This file chooser prompts for a directory to write the files to and a prefix to name the files with. Press `Save`.
    - View the `Details` tab in the Properties dialog again.
3. Save remote data into a bundle.
    - Load in the `Davenport Radar` favorite again.
    - From the **Quicklinks History** page select the RUC data source.
    - Save the bundle as a `.zidv` file again. Select both data sources to save.
    - Open the bundle.
4. How do I get at the data files?
    - A `.zidv` file is just a zip file.
    - From a terminal window do:

    ```
    unzip <your zidv file>
    ```

3.6.5 Image Capture

In this section, we will cover how to capture images within the IDV.

1. If you have nothing displayed, then load in the bundle that we previously created.
   - Select the **File** **Open** menu item. In the `File Dialog` choose `workshop.xidv` from your home directory.
2. Select the **View** **Capture** **Image...** menu item from the map view's menu bar. The `File Dialog` will popup.
3. The IDV requires that the window you are capturing an image in is not obscured. Move any other windows out of the way.
4. You can save an image as a jpg, gif or png. Make sure you are in the directory `/home/idv`. Enter the name `workshop.jpg`. Click `Save`.
5. Let's capture the full window. From the File Dialog enter `fullwindow.png` and select the `Full Window` button on the left.
6. The IDV treats image files as yet another data source. Load in the newly created images by:
   - Open the `Data Source Chooser`.
   - Go to the `Files` tab.
   - Make sure you are in your home directory.
   - Select `workshop.jpg` and `wholewindow.png` with a control-click
7. Since images are not geo-located we can only display the image in the `Omni Control`. Create this display.
8. Now, let's save this image as a Google Earth kmz file.
   - When we capture an image for Google Earth we have to make sure we are using a Lat/Lon projection in an overhead view.
   - The easiest way to do that is to select the **Projections** **Use Displayed Area** menu.
   - Select the **View** **Capture** **Image** menu item and enter `workshop.kmz`.
   - A reminder is shown about the Lat/Lon projection.
   - From a terminal window bring up Google Earth and Open this kmz file.
9. The IDV can also read in Google Earth kml/kmz files.
   - From the IDV Data Source Chooser select the `workshop.kmz` file.
   - From the field selector create an Image display.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.6.6 PDF/PS/SVG Capture

In this section we will cover how to capture PDF, Postscript and SVG images within the IDV.

1. Clear your displays and create a contour plan view.
2. Select the **View Capture Image...** menu item from the map view's menu bar. The `File Dialog` will popup.
3. You can save the display as a PDF, Postscript or Scalable Vector Graphics file. Make sure you are in the directory `/home/idv`. Enter the name `workshop.pdf`. Click `Save`.
4. In the dialog that pops up select `Preview` and also enter some label text.
5. On the workshop Linux machines run *evince* to view the PDF

   ```
   evince workshop.pdf
   ```

   ♦ Vector capture is a 2 stage process. The raster-like displays (e.g., images) are captured then the vector displays (e.g., contours, maps).
6. Lets look at the limitations of vector catpure.
   ♦ Leaving the contour plan view also create an isosurface display.
   ♦ Make sure the contour plan view intersects the isosurface at some level.
   ♦ Capture and view `workshop2.pdf`.

3.6.7 Movie Capture

In this section we will cover how to create QuickTime movies, animated gifs and Goole Earth files within the IDV.

1. First, load in some data with multiple time steps and create a display.
    - Remove any data and displays.
    - Load in the `ETA 1998-06-29 00:00` data source.
    - Create a `Contour Plan View` display with a `temperature` parameter.
2. Save this state off as the "movie" bundle.
    - Select the **File Save As...** menu item.
    - Make sure you are in your home directory and enter `movie.xidv`.
3. Open the `Movie Capture` window by selecting the **View Capture Movie** menu item from the map view's menu bar.
4. For now the IDV requires that the window you are capturing an image in is not obscured. Move any other windows out of the way.
5. A movie is simply a sequence of frames. There are three ways to capture this sequence.
6. You can capture one frame with the `Capture one image` button.
    - Try 3 or 4 cycles of capturing one image and moving the viewpoint.
    - You should see the `Frames` panel become enabled and the number of frames listed.
    - Bring up the `Movie Preview` window by clicking the `Preview` button.
    - In the `Movie Preview` window you can step forward, backward and play the movie as well as delete individual frames.
7. Start fresh by clicking the `Delete all` button in the main `Movie Capture` window.
8. Now, let's capture an entire animation by clicking the `Capture Time Animation` button.
9. When the animation is complete the `File Dialog` will automatically be shown. Make sure you are in the directory `/home/idv`. Enter the name `workshop.mov`. Click `Save`.
10. The `Capture automatically` button allows you to capture a single frame every *N* seconds.
    - Click the `Capture automatically` button to start the capture.
    - Turn on animation.
    - Navigate around the 3D view. Zooming in, etc.
    - Click the `Stop` button and then click `Preview`.
    - The `Preview` window allows you to view, play and delete the frames that make up a movie.
11. The IDV treats QuickTime movies as yet another data source. Load in and view the newly created movie by:
    - Open the `Data Source Chooser`.
    - Go to the `Files` tab.
    - Select `workshop.mov`.
    - For Quicktime movies the data source is not shown and the `Movie Display Control` will automatically be created.
12. You can also save the movie as a Google Earth kmz file.
    - Press the `Delete All` button.
    - Capture the animation with the `Capture animation` button.
    - When done, choose `movie.kmz` as the file.
    - Bring this file up in Google Earth.

Footnotes:
To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

To load in the `ETA 1998-06-29 00:00` data source:

- Open the `Catalog Chooser` by either:
  - Choose the **Data Choose Data From a Catalog** menu item.
  - Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Case Studies Data for Comet Case Study 039 NCEP Model Data ETA 1998-06-29 00:00 GMT** item.
- Select the `Add Source` button.

To create a `Contour Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature` field.
- In the `Displays` list, select `Contour Plan View` and press the `Create Display` button.

Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

3.6.8 Saving Data

In this section, we will cover how to write out point and gridded data files.

1. First, clear all existing data and displays and load in the Sample ETA Grid from the catalog chooser.
2. Create a Contour Plan View of Temperature @ 500 MB.
3. Save the displayed data.
   - In the Dashboard Displays tab, select the **File Save Export Displayed Data to NetCDF...** menu from the display's File menu.
   - Enter `testgrid.nc`.
4. Now, let's load this grid back into the IDV. Note the parameter differences. Create another Contour Plan View with T. **Data Choose Data From the File System**
5. Let's save a derived quantity.
   - Choose Geostrophic Wind Vectors (from Z) under 3D Grid - Derived and create a Vector Plan View.
   - Repeat the above steps, saving the file as testgrid2.nc
6. Let's save point data
   - Load in a couple of time steps of real time METARS from the ADDE Point Data chooser.
   - Create a Station Model display of the data.
   - Write out the NetCDF file with the **File Save Export to NetCDF** menu.

3.6.9 RAMADDA

In this section we will cover how the IDV can publish images, movies, bundles and data to a RAMADDA server. RAMADDA is a content management system

- First we need to install the RAMADDA Publisher plugin.
    - Go to the **Tools Plugin Manager** menu to bring up the Plugin Manager.
    - Under Miscellaneous install the RAMADDA Publisher
    - Shut down and restart the IDV.
- Create a publisher account.
    - Go to the **File Publish New RAMADDA Publisher** menu.
    - Enter:
      Name: *Workshop account*
      Server: *motherlode.ucar.edu*
      User Name: *workshop*
      Password: *unidata*
    - Press OK.
- Now, any of the file dialogs for saving images, movies, data or bundles will have a "Select Publisher" menu.
- Publish an image and bundle to RAMADDA
    - Load up some data and create a display.
    - Save an image off under the **View Capture Image** menu.
    - In the File dialog enter a image file name and also select the "Workshop account" to publish to.
    - In the Publisher dialog enter a name, description and select **Top IDV Community Resources Workshop Sandbox**
- In the Catalog chooser go to **Unidata's RAMADDA Server IDV Community Resources Workshop Sandbox**. Note - If you right click on the entry you can Load Remote Catalog and then do a refresh to update for any changes.
- Select one of the bundles and load it into the IDV.

3.7 Point Observations

The IDV can display a variety of point observations

3.7.0 Loading Surface Data

METAR and Synoptic data can be accessed in the IDV from ADDE servers. The data available is hourly, each hourly file having all station observations at that time.

1. If other displays and data are already loaded, select the **Edit** **Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. In the `Data Source Chooser` go to the `Point` tab.
3. In the `Server:` entry box, select the remote ADDE server "adde.ucar.edu".
4. In the `Data Type:` entry box, select the choice "Surface (METAR) Data" and click on `Connect`.
5. The IDV will connect to the ADDE server at adde.ucar.edu and download metadata about the available METAR data.
6. Select `Absolute times` to see a list of data times that you can get from this server.
7. The bottom time in the list, the most recent time, should be highlighted. If it is not, click on it.
8. Hold down the control key and click on the second to last time so that two times are selected.
9. You could also select `Use relative times` and select `Most recent 2 times`.
10. Make sure the `Layout Model:` is "Observations>METAR".
11. Click the `Add Source` button.
12. Because the `Create Display` checkbox is selected the IDV automatically creates a point data plot.

3.7.1 Surface Point Observation Displays

The IDV surface point observation display makes weather station plots.
If necessary, select surface data sources and times as described in (see Loading Surface Data).

1. If you do not have a Surface Plot created then create one from the Field Selector.
2. Zoom in and see more stations
   - Zoom and pan as needed to fill the display with Colorado. As you zoom in you will see more stations appear in the plot.
   - This display's control window appeared when the display was made. You can control the density of the stations plotted using the `Plot Density` slider.
   - To see all stations, click on the `Declutter` checkbox to toggle it off.
   - In the control window, use the `Layout Model` selector box to change the layout model from METAR to Temperature. Later in the workshop you learn how to change and create layout models. (See the Layout Model Editor).
3. Changing Z position.
   - Rotate the view to see that the stations are displayed at their actual altitude.
   - For the `Vertical Position` select `Fixed position` on the `Point Data Plot Control` to display the stations at a fixed level.
4. We're going to find out where its hot and humid.
   - Select the `Filters` tab on the `Point Data Plot Control`.
   - In the first row under `Property` select `T`.
   - Select ">".
   - Under `Value` enter "80[F]". The "[F]" is an optional unit.
   - In the second row under `Property` select `TD` - dewpoint temperature
   - Select ">".
   - Under `Value` enter "70[F]".
   - Press the `Apply Filters` button.
   - Click on one of the stations to see the full listing in the `Data Readout` tab.
5. Remove All Displays
   - If other displays are in the main IDV window, select the **Edit Remove All Displays** menu item or the icon in the toolbar to clear them out.

3.7.2 Subsetting Point Data

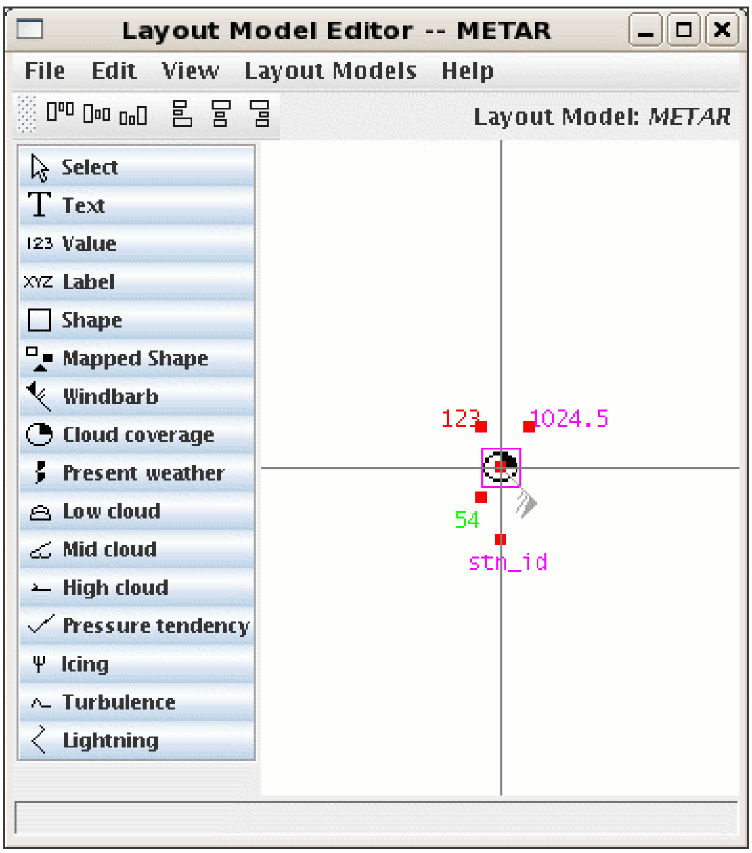
You can do time binning and spatial subsetting for point data.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load in the file `/data/idv/point/madis.nc`. Make sure you the Data Source type select "netCDF Point Data files"
3. Create a Point Data Plot. Say Yes to seeing all of the times. Notice, the number of times. Not very workable.
4. In the display control window go to the `Times` tab. Turn on the `enabled` button and enter 5 minutes for the `Only Show Every`. Press return.
5. Note: this isn't time binning. This just skips the obs.
6. Remove that display.
7. Now we're going to bin the times.
    - Bring up the Properties dialog for this madis point data. (Double click or right click on the data source entry in the Field Selector).
    - Choose a 15 minute `Bin Size` and press `OK`.
    - Now, create a point data plot again.
8. Spatially subset the point data.
    - Bring up the Properties dialog for the madis point data. Go to the `Spatial Subset` tab.
    - Zoom in over Colorado and drag a box with the left mouse to select a region.
    - Turn on `Reload Displays` and press `OK`.

3.7.3 Layout Model Editor

The Layout Model Editor allows you to view, change and create Layout Models.

1. First, load in the layout model example from the **Toolbar** **Workshop** **Station model exercise** menu.
2. Open up the `Layout Model Editor` with the edit layout model button: in the display control window or from the **Tools** **Layout Models** menu.
3. You should see something like this:



4. The `Layout Model Editor` consists of:
   - ♦ A drawing canvas.
   - ♦ A list of symbols.
     - ◊ `Select` Is an editing command. Puts the editor into select mode.
     - ◊ `Text` Displays text. Used for station identifiers, text results from Jython, etc.
     - ◊ `Value` Displays a numeric value. Lets you specify a display unit and a format.
     - ◊ `Label` Shows a fixed text label.
     - ◊ `Shape` Shows a shape.
     - ◊ `Mapped Shape` Maps a value to a shape.
     - ◊ `Windbarb` Shows wind direction and speed.
     - ◊ `Cloud coverage`, `Present weather`, `Low cloud`, etc. These symbols are used to map a value into a predefined range of display symbols.
   - ♦ An alignment toolbar.
   - ♦ **File**, **Edit**, **View**, **Layout Models** and **Help** menus.

5. Let's go look at the `METAR` layout model.
    - Right click on the symbol in the upper left that represents temperature and select **Properties**
    - Change some of the properties (e.g., scale, background color, format) and click `Apply`.
    - The `Apply` and `OK` buttons save your changes to the layout model in the editor.
    - The `Save` button saves the changes plus writes out the layout model and updates the display.
    - Close the dialog and use the **File** **Remove** menu to get rid of your local changes. Re-select the `METAR` layout model with the **Layout Models** menu.
    - Try moving some the position of the symbols around. To apply the change to the display use the **File** **Save** menu item.
    - Look at the `Earthquake` layout model. View the properties for the symbol.
6. Create a new Layout Model, called "Workshop".
    - Select the **File** **New** menu item.
    - In the dialog box enter the name of the new layout model, "Workshop".
    - Click the `OK` button.
7. Add a symbol.
    - Select the `Text` symbol in the symbol list.
    - Click in the center of the drawing canvas.
    - In the `Properties` dialog you can select what parameter from the data set is used and other display information.
        - The value in the `Parameter` field should be "IDN", the WMO Station ID.
        - Select *Courier* in the **Font** menu.
        - Select an 11 point font in the **Size** menu.
        - Click the `OK` button.
8. Position the symbol.
    - Click and drag the symbol to the center of the screen.
    - Note the red dot on the symbol. This is the *Alignment point*. It determines the point on the bounding box of the symbol that is used to align the symbol with the center point of the display coordinate space.
    - Right click on the symbol to bring up the **Symbol Menu**
    - Select the **Alignment Point** **NW** menu item. This should change the red dot on the symbol to the upper left corner.
    - Right click on the symbol again, but this time select the **Center** **Center** menu item. This aligns the symbol's alignment point to the center of the display space.
9. Save this layout model with the **File** **Save** menu item.
10. Close the window by selecting the **File** **Close** menu item.
11. Back to the `Point Data Display`, choose the `Workshop` layout model. Humm, why isn't anything displayed?

3.7.4 More on the Layout Model Editor

Let's find out why there isn't anything displayed.

1. Bring up the `Workshop` layout model again.
    - From the `Point Data Plot` display control window click on the edit image: .
    - Bring up the `Properties` dialog for the symbol by right clicking on it and selecting the **Properties...** menu item.
    - Note that the name of the parameter we are using is "IDN".
    - Press the `Cancel` button to close the dialog.
2. Now, back to the `Point Data Plot` control window let's see what parameters are available in the data set.
    - Normally we would do the following but the current version of the IDV does not support it. Select the **Help Details...** menu item to bring up the `Display Control Details` window. This lists the fields available in the data set.
    - So instead, we will look at `Details` tab in the `Properties Dialog` for the Point Data Source.
    - Note there is no `IDN` field. However, there is an `ID` field in the data.
    - Close the `Properties` window.
3. Once again, back to the `Layout Model Editor`.
    - Bring up the `Properties` dialog for the symbol.
    - Click on the button to bring up the list of all parameter names. Look for and select the "ID" parameter under the point data.

      <u>A note on the data source name</u>
    - Aha! We were using the `IDN` when we should have been using the `ID`, the IACO Station ID.
    - Select that parameter. Notice that the parameters can be a comma separated list. Click the `OK` button and save the layout model (select **File Save**).
    - You should now see the station identifiers in the display.
4. Now, how do we handle the case where we have lots of layout models for data that has "ID" like fields and then we encounter data with some other field name, e.g., "the_station_id"? Do we go and change each and every model?
5. Now, let's spend some time in building some other layout models.
    - Start fresh.
    - Go to the CONUS projection with the **Projections Predefined US CONUS** menu item.
    - Load in US city locations with the **Displays Locations US U.S. Cities** menu item.
    - Select the `Layout Model` button.
    - Edit the `Location` layout model.
    - Try adding a `Shape` shape.
    - Select some shape and press `Apply`
    - The location has a `pop` attribute. Try scaling the shape this value.
        - Go to the `Scale Size By` tab in the `Properties Dialog`
        - Enter `pop` for the parameter.
        - In `Data Range` enter 1000000 in the second field.
        - Enter some scale range. e.g., 0.5-5.0
        - Press `Apply`
    - Try doing a color by `pop`

Footnotes:
As an aside notice the really long name for the data source. Its too late to change it in this menu but we could change the name in the `Properties` dialog for the data source.

To start fresh choose the **Edit** **Remove All Displays and Data** menu item or the scissors icon in the toolbar.

3.7.5 Observation List Display

The Observation List Display Control shows a textual listing of observation data. We are going to look at some earthquake point data.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load some point data.
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `Files` tab.
   - Go to the point data directory: `/data/idv/point/`.
   - Select `earthquake.nc` (Note: **not** `earthquake.csv`)
   - Under the `Data Source Type` pulldown menu, select "Netcdf Point Data files"
   - Press `Add Source`
3. Create the display
   - In the `Fields` panel select `Point Data`.
   - In the `Displays` select `Point Data List` and press `Create Display`
4. Modify the displayed fields.
   - Press the `Select Fields` button
   - We see there are 4 fields shown and one field (`var`) not shown.
   - Select `var` and press `Add`
   - Change the order of the fields - select `var` and then press the up arrow button.
   - Press `OK`.
5. Modify the listing.
   - Toggle `Show Raw Data`
   - Sort the columns by pressing the `var` column header.
   - Figure out what is the maximum depth of the observations.
6. Export the data as a text CSV file.
   - Use **File Save Export Table** in the `Point Data List` display control.
   - Enter `earthquake.csv`
   - Read in this file from the File Chooser. Make sure you set the `Data Source Type` to "I'm Feeling Lucky"
   - Create another `Point Data List` display with this CSV data.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

3.7.6 More with Point Displays

Leave the earthquake point data source and list display.

1. Create a `Point Data Plot` display
   - In the `Fields` panel select `Point Data`.
   - In the `Displays` select `Point Data Plot` and press `Create Display`
2. If the `Earthquake` station model is not selected for display then select it with the `Change` button. This station model scales by the `var` field and colors by `Altitude`.
3. Turn on the bounding box if it is not on with the **View Show Wireframe Box** menu.
4. Tilt the view to be looking from the south. Note the map is at the bottom of the box and the earthquake symbols are below that.
5. Change the vertical scale of the display.
   - Find the altitude (depth) range of the data from the Observation List Display created in the prior exercise.
   - Select the **View Viewpoint Vertical Scale** menu.
   - Enter the min value found in the Observation List display.
   - Enter 0 for the max value.
   - Press `Apply`
   - Note the map is now at the bottom. Raise the map by bringing up the `Background Maps` display control window, select the `Display` tab and use the `Map Position` slider, close the window.
   - If happy with the vertical scale press `OK` button.
6. Note the gray scale on the default background is difficult to see. Change the background color.
   - Select the **View Color Set Colors** menu item.
   - Press the `Set` button in the background row.
   - Choose an appropriate color (e.g., light blue) and press `OK`
   - Press the `Apply` button. If things look ok press `OK`
7. Figure out what depth range contains most of the earthquakes.
   - Rotate to a southern view. You may need to disable `Declutter`.
   - Add a `Location Indicator` display with the **Displays Special Location Indicator** menu.
   - Move the origin point with the left mouse button.
   - Move the bearing point with a control-key/left mouse button.
   - Look at the altitude readouts.
   - Go to the `Display` tab and turn on `Solid:X` and `Solid:Y` checkboxes.
   - Rotate the display and drag the origin point up and down.
   - Change the `Transparency` to 50%

3.7.7 Exercise: Japan EarthQuakes Display

We are going to look at another earthquake display combining Japan historical earthquakes and 2011 earthquakes point data in this section.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. First create two layout models for this exercise.
    - Open up the `Layout Model Editor` from the **Tools Layout Models** menu.
    - Create a `Shape Symbol` with the shape as `Sphere`, double click the symbol to bring up the `Properties Dialog` and set the Foreground Color as blue and Scale Size 0.2, and save as `EarthQuake1`.
    - Create another `Sphere` shape, with `Foreground Color` set as Clear, `Scale Size By` 0.2, in the `Color By` tab, set `Map Value of` as `magnitude`, and set `Data Range` 0 and 9, in the `Scale Size` tab, set `Scale By Parameter` as `magnitude`, set `Data Range` 0 and 9, and set `Scale By Factor` 0.3 and 3. Now save this model as `EarthQuake2`.
3. Load Japan historical earthquake point data.
    - Open the `Data Source Chooser`.
    - In the `Data Source Chooser` click on the `Files` tab.
    - Go to the point data directory: `/data/idv/point/`.
    - Select `IEB-Sendai-region-3000quakes-before-1Mar2011-mags-ge-5.nc`
    - Under the `Data Source Type` pulldown menu, select "Netcdf Point Data files"
    - Press `Add Source`
    - Bring up the `Properties` window, and set the `Bin Size` to 5 years.
    - In the `Fields` panel select `Point Data`.
    - Select the `Layout Model` `EarthQuake1`.
    - In the `Displays` select `Point Data List` and press `Create Display`
4. Modify the displayed setting.
    - In the `Displays` control window, under the Layout tab, uncheck the `Declutter`.
    - Select the `Times` tab, click the `Multiple`, and change the `Range` to be `Start from data` and `End from animation`.
5. Load 2011 Japan earthquake point data .
    - Load another dataset in the directory `IRIS-EB-quakes-Japan-01Mar2011-1201-16Mar2011-530quakes.nc`.
    - Set the `Bin Size` to be 5 hours.
    - Select the `Layout Model` `EarthQuake2`.
    - Create Display and repeat the display setting in the previous step.
6. Start the Animation.

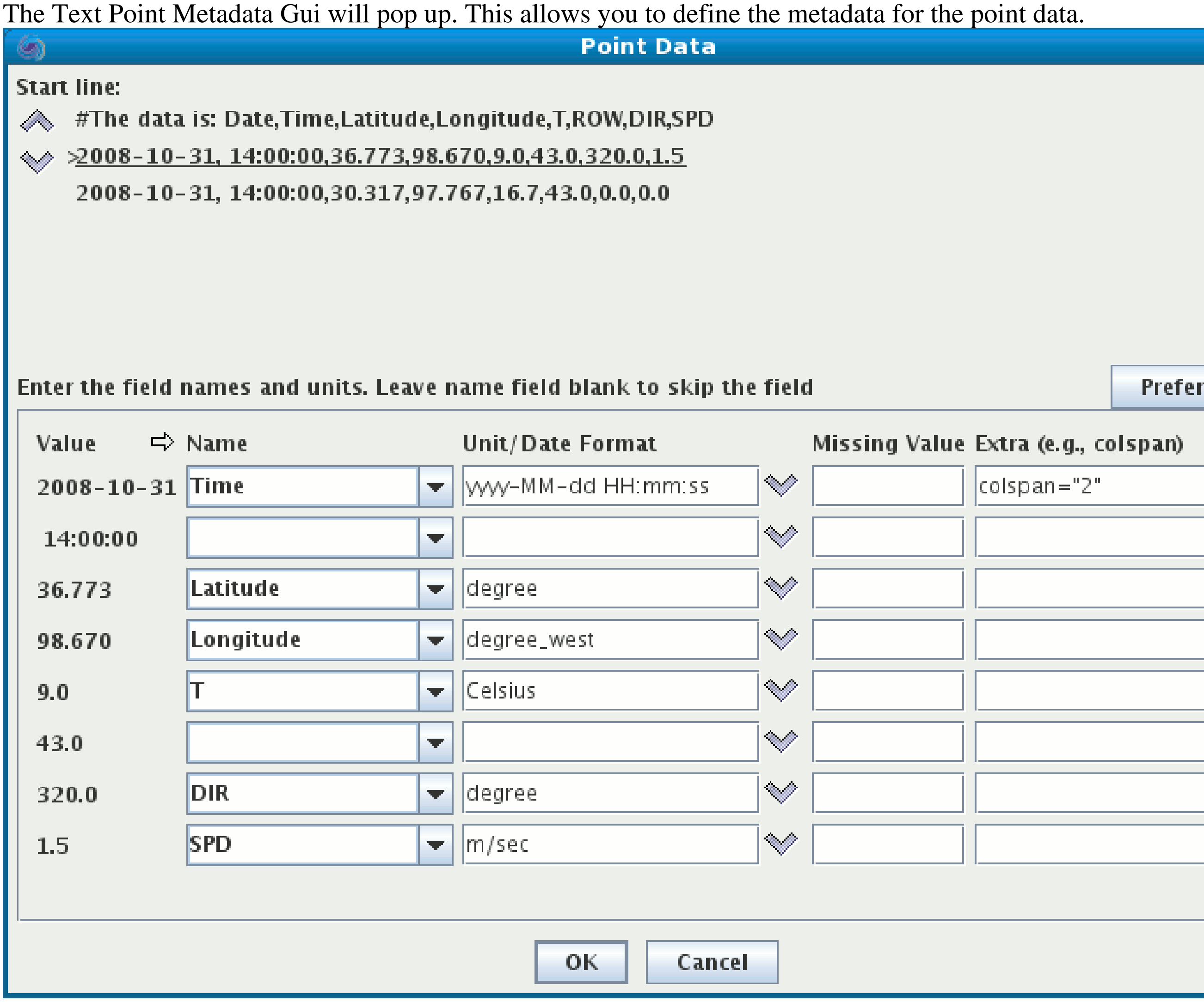
Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

3.7.8 Text (ASCII) Point Data

The IDV can read in point observation data in Comma Separated Value (CSV) form (as well as tab separated values and Excel xls files).

- Here is an example file: metar.csv. The minimum data required by the IDV is time, latitude and longitude.
- Load the csv file in.
    - In the `Data Source Chooser` click on the `Files` tab.
    - Go to the point data directory: `/data/idv/point/`.
    - Select `metar.csv` and press `Add Source`.
- The Text Point Metadata Gui will pop up. This allows you to define the metadata for the point data.



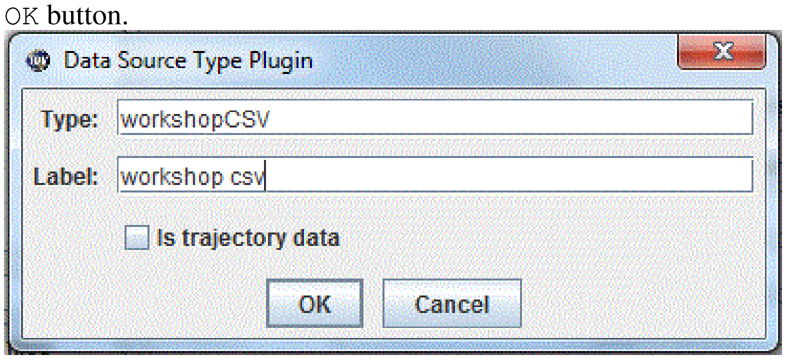
- Using the Text Point Metadata Gui
    - At the top a number of the initial lines from the text data are shown. The arrow buttons allow you to specify where the data starts. Skip the first line.
    - For each column of text data there is a row that shows sample value and allows you to enter a name, unit, date format, missing value and extra information.

    - We're going to skip the "ROW" parameter and define metadata for: "Time ","Latitude", "Longitude", "T", "DIR",and "SPD".
    - Note that the time field spans two columns - year and time. The IDV needs to have the time in one column so we use:
      colspan="2"
      This tells the IDV to concatenate columns 1 and 2 and treat them as a single column.
    - Note: the unit for longitude is "degrees_west"
    - Enter the names and units as shown above.
    - You can save these settings for future use.
        - Press the `Preferences` button and select **Save Current**.
        - Enter a name, e.g., "workshop"
        - You can reapply these settings with the `Preferences` button.
    - When done press `OK`. You can also bring up this GUI in the Data Source Properties Dialog.
- Create the `Point Data Plot` and change its layout model to "METAR"
- You can also add a special 2 line header to the top of your text file.

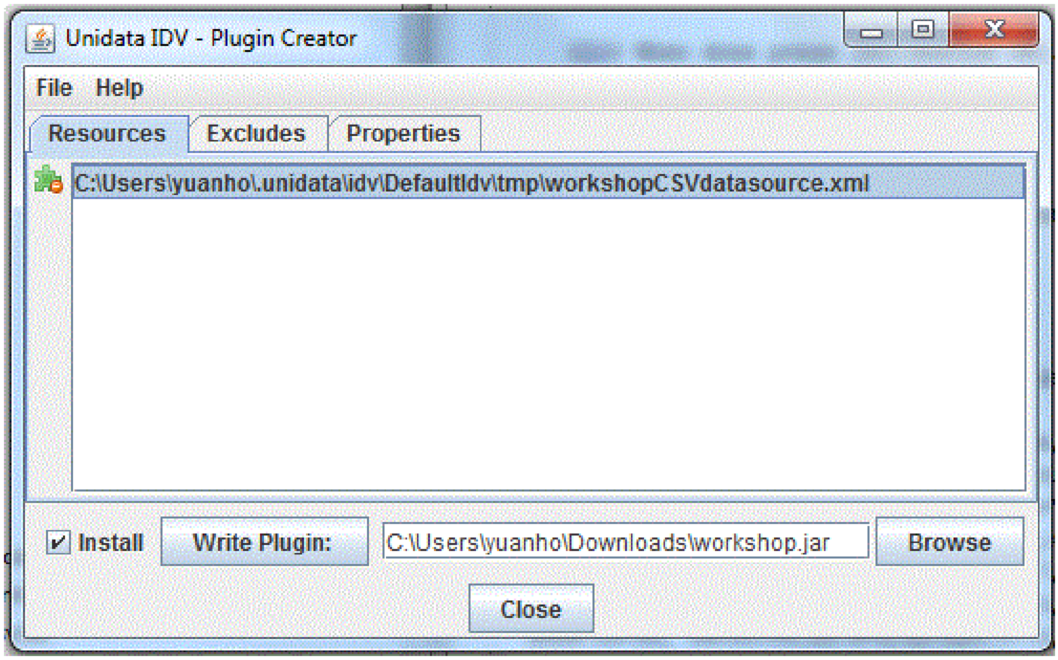
```
(index) -> (Date,Time,Latitude,Longitude,T,DIR,SPD)
 Time[fmt="yyyy-MM-dd HH:mm:ss"],Latitude[unit="deg"],Longitude[unit="degrees west"],T[un
       skip,DIR[unit="deg"],SPD[unit="m/s"]
 2008-10-31, 14:00:00,36.773,98.670,9.0,43.0,320.0,1.5
 2008-10-31, 14:00:00,30.317,97.767,16.7,43.0,0.0,0.0
 2008-10-31, 14:00:00,34.398,96.148,13.0,43.0,0.0,0.0
 2008-10-31, 14:00:00,28.973,95.863,17.0,43.0,60.0,1.5
 2008-10-31, 14:00:00,45.150,89.150,8.0,43.0,310.0,2.6
          .......
```

In the example above, the first line's structure is a VisAD type definition. The second line defines the formatting and units of the parameters. Starting the third line is the data section of this sample. There is more documentation in the User Guide.

- You can also create the data source plugin for your text file.
    - Press the `Preferences` button and select **Write Data Source Plugin**.
    - Enter a name of type, e.g., "workshopCSV", and a label, e.g., "workshop csv", and click the `OK` button.



    - Enter a plugin name, e.g., "workshop.jar",and check the install box, and click the `Write Plugin` button.



- When the IDV is restarted, the data type "workshop csv" is showing in the list.

3.7.9 Objective Analysis of Point Data

The IDV can create gridded data fields from point data using Barnes Objective Analysis.

- Leave the metar.csv point data source from the prior exercise.
- In the Field Selector open the "Gridded Fields" tab. This lists the numeric fields in the data.
- Choose "T" and create a `Contour Plan View` display.
- In the `Grid Parameters` tab check off `Use Default` and change the number of passes to 6 and create another `Contour Plan View` display.
- Toggle the visibility to compare the differences.
- Create a `Point Data Plot` display and change its Layout Model to "Temperature" and compare the values to the grid values.
- Change other values and create display to see what effect they have on the resulting grid.

3.7.10 Doing More with Point Observation data

1. Create a plot of Surface (METAR) data for 12Z today. Overlay the contours of mean sea level pressure for the same time from the NAM model.
2. Load in the latest 12Z RUC model data. Create a `Color-Filled Contour Plan View` display of `Temperature at fixed height above ground` for the initial time step. Overlay regular contours of the same field in black. Create a plot of the 12Z surface data for the same time with a `Temperature` station model. How do the grid and observations compare?

3.7.11 Lightning Display

The IDV has lightning data display capability.

1. If other displays and data are already loaded, select the **Edit** **Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load some lighting data
    - Open the `Data Source Chooser`.
    - In the `Data Source Chooser` click on the `Files` tab.
    - Go to the point data directory: `/data/idv/lightning/`.
    - Select both `201118719.ingest` and `201118720.ingest`
    - Under the `Data Source Type` pulldown menu, select "Netcdf Point Data files"
    - Press `Add Source`
3. Create the display
    - In the `Data Sources` panel right click the lightning data source.
    - Under `Properties` tab in the Time Binning section, change Bin Size to 5 minutes
    - Click on the `Spatial Subset` tab. Select a region encompassing South Carolinas state and press `OK`.
    - In the `Fields` panel select `Point Data`.
    - In the `Displays` select `Point Data Plot` and press `Create Display`
    - Select `Layout Model` tab
    - Click on the menu and select Lightning > Bolts
    - In the same menu click Edit
    - Double click on the lightning layout model
    - Ensure that the parameter being displayed is "sgnl", and Press `OK`.
    - Press `Create Display`
4. Improving the lightning display in the `Displays` panel
    - Disable `Declutter`.
    - In the `Times` panel, select `Show Multiple`
    - Select the button to show the `Time Settings` dialog
    - For Start Time choose `Relative to End Time`, and click on button to the right of that dropdown
    - In the `Set Start Offset` Dialog select minus (-) 20 minutes.
    - For End Time choose `From Animation Time`, and Press `OK`
5. Turn on animation

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.7.12 Overlaying Lightning Data on Radar Data

Overlaying lightning data on radar data.

1. Load some radar data
    - Open the `Data Source Chooser`.
    - In the `Data Source Chooser` click on the `Files` tab.
    - Go to the radar data directory: `/data/idv/lightning/`.
    - Select all the `*.nids` files
    - Under the `Data Source Type` pulldown menu, select "Radar files (NEXRAD/DORADE/UF)"
    - Press `Add Source`
2. Create the display
    - In the `Fields` panel select `Reflectivity`.
    - In the `Displays` select `Radar Sweep View in 2D` and press `Create Display`
    - Press `Create Display`
3. As an exercise for the reader, turn on various geographic data sets such as US State lines.
4. Turn on animation
5. Where are strikes occurring relative to areas of high reflectivity?

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

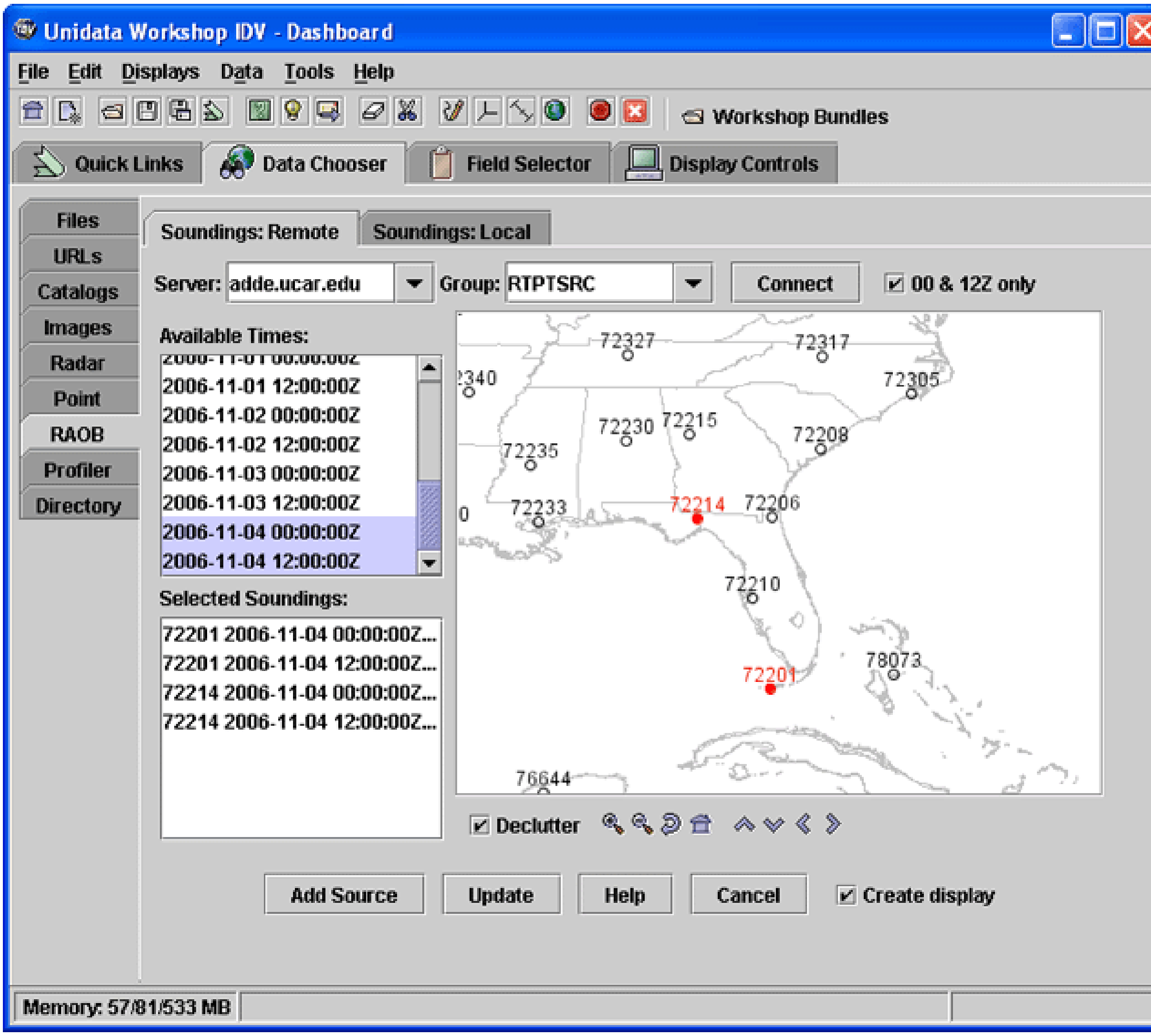
3.8 Upper Air Displays

The IDV displays upper air data. Presently the IDV displays balloon soundings (RAOBs) as a skew-T log-p diagram, and with a wind barb staff and a table of aerological values.

3.8.0 Sounding Displays

Upper Air RAOB sounding data can be accessed and displayed in the IDV as skew-T log-p, Stuve or Emagram displays, with a table of aerological values. More than one station and more than one sounding at each station can be loaded at once.

1. Open the RAOB Data Source Chooser and Select a Time
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `RAOB` tab.
   - Click on the `Soundings: Remote` tab.
   - In the `Server:` entry box, select the remote ADDE server `adde.ucar.edu`. `Group:` entry box, select `RTPTSRC`. Make sure the `00 & 12Z only` box is checked
   - Click on `Connect`.
   - The IDV will connect to the ADDE server at adde.ucar.edu and download metadata about available RAOB data. Data times you can get from this server are listed in the box `Available Times:`.
   - Select the 00 and 12 times for today.
2. Zooming and Decluttering the Selector, and Selecting Stations
   - The map only shows a few stations, so you need to zoom in and declutter.  More stations will appear when you zoom in and declutter. Find station 72201 (Key West) southwest of Miami, Florida.
   - Click on that station. It will turn red, and it will be listed with the data times in the `Selected Soundings:` box.
   - Select another station, 72214, north of 72201, by holding down the Ctrl key and clicking on 72214. Both stations now should be red.

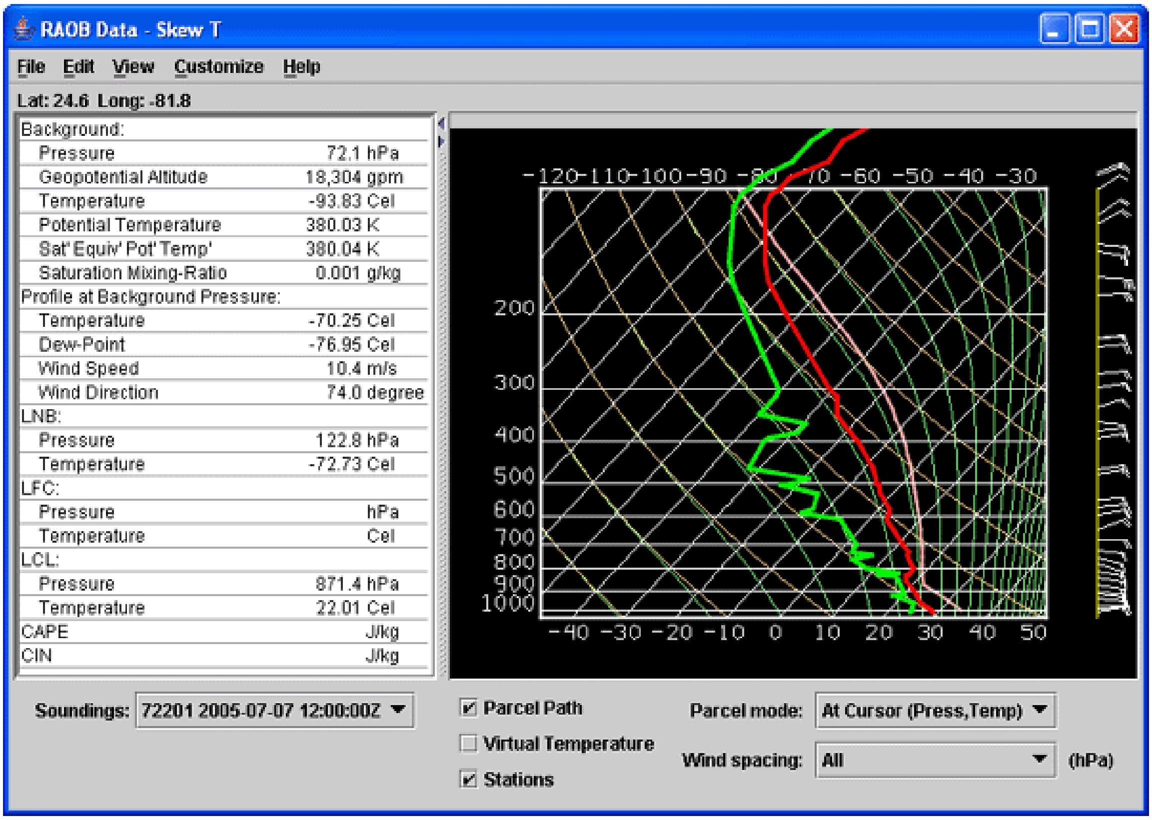


- Four entries should be in the `Selected Soundings:` box. Click the `Add Source` button at the bottom of the selector window.

You should see the label `RAOB data: adde.ucar.edu` in the `Data Sources` panel in the `Field Selector`. If the `Create display` checkbox was selected in the chooser, the data will be read from the server and a `Skew-T` display will be created. If not then create it:

- In the `Field Selector` window create a `RAOB Skew-T` display using the `RAOB Data` field.

The skew-T plot appears in its own control window in the `Dashboard`, with the data from the first station and time plotted. Skew-T plots of RAOB data have a station selector box labeled `Soundings`. Undock the `Skew-T` control from the Dashboard with the **View Undock from Dashboard** menu.



- Using the Skew-T Control
  - The skew-T plot, wind barb staff, and associated aerological table are all in the control window.
  - Move the mouse pointer over the skew-T Plot and see the values on the plot listed in the table.
  - Use the `Soundings` selector box to change station and time selection.
  - Use the **Customize** **Display Types** **Stuve** menu item to change the display to a Stuve.
  - Show the idealized parcel path by checking the `Parcel Path` checkbox. Change the `Parcel mode:` to the `At Cursor Pressure` option. Click in the display with the middle mouse button to show the parcel path from that pressure.
  - Modify the sounding by clicking and dragging on the temperature profile. Change the `Parcel mode:` to the `At Cursor (Press,Temp)` option. Click in the display with the middle mouse button to show the parcel from new locations on the modified sounding. For more information, see Using the IDV Sounding Displays
  - Reset the profile with the **Edit** **Reset Sounding** menu.
- Remove All Displays and Data
  - If other displays and data are already loaded, select the **Edit** **Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.

Footnotes:
Open the `Data Source Chooser` by either:

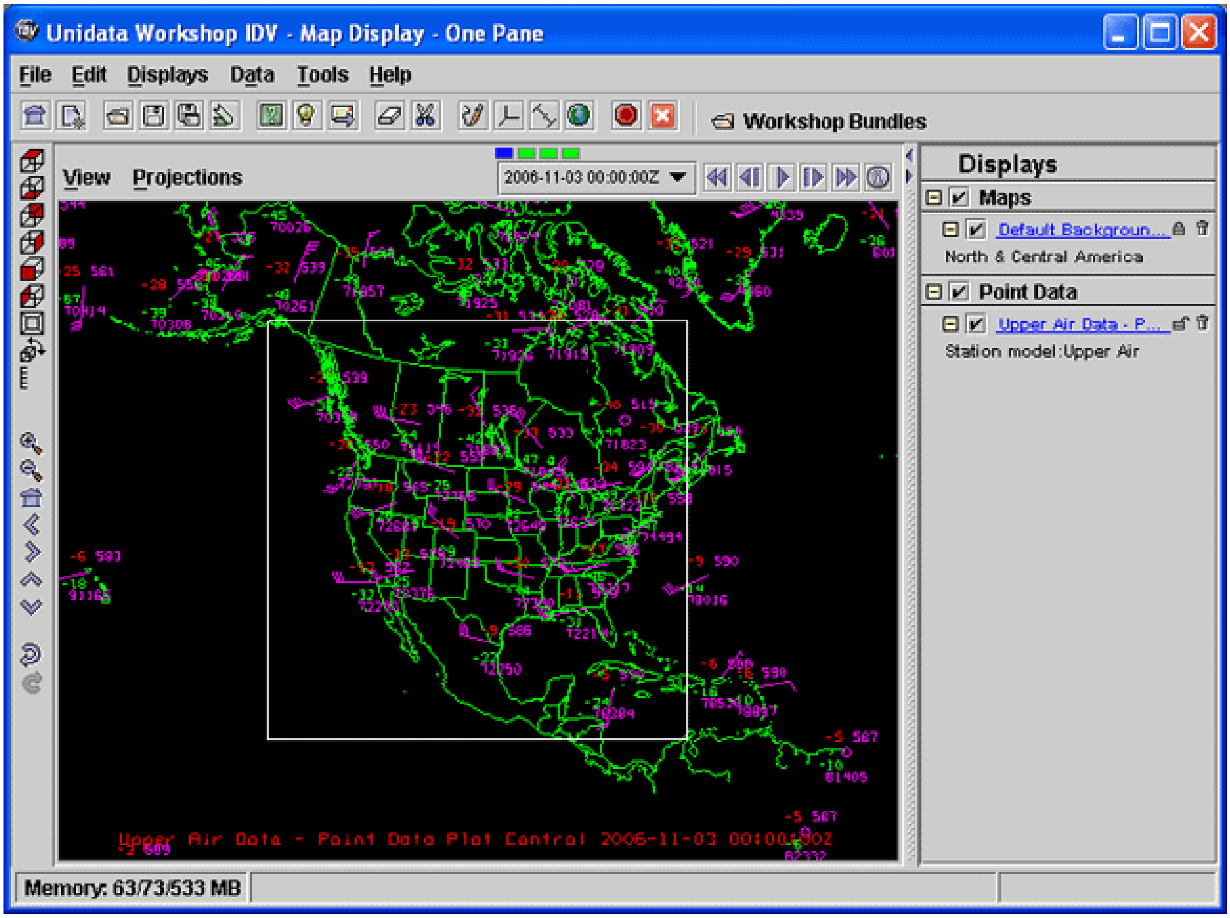
- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.8.1 Single Level Point Data Displays

The IDV can plot upper level maps as point displays from the RAOB data.

1. Open the RAOB Single Level Data Source Chooser
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `Point` tab.
   - Click on the `Upper Air` tab.
   - In the `Server:` entry box, select the remote ADDE server `adde.ucar.edu`. In the `Data Type:` entry box, select the `Upper Air Data` entry.
   - Click on `Connect`.
   - The IDV will connect to the ADDE server at adde.ucar.edu and download metadata about available RAOB data. You can load data in by relative times, or by absolute times. Select the `Relative times` radio button and the `4 most recent` times option.
   - From the `Station Model` list, select `Upper Air`.
   - From the `Level` list, select `500` for the 500 millibar level.
   - Click the `Add Source` button.
2. If the `Create display` option was checked in the chooser, a `Point Data Plot` will automatically be created. If not, create one.
   - In the `Field Selector` window create a `Point Data Plot` display using the `RAOB Point Data` field.

   The point data plot appears in the main window.



3. Set the projection to the continental U.S. (**Projections Predefined US CONUS** menu).
4. Use the `Point Data Plot` control to change the plot density to show more stations.
5. Remove the display
6. Now, plot data at a different level and region.
   - In the `Field Selector` window select a different level (e.g. 250 mbar) in the `Level` tab of the `Data Subset` panel.
   - Under the `Region` tab, select a region of interest.
   - Click the `Create Display` button.
8. Remove All Displays and Data

   - If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

3.8.2 Doing More with Upper Air (RAOB) Displays

1. Load in the `Davenport Radar` bundle from the **Workshop** favorites. Use the `CCS039` group in the RAOB chooser to select data from the Comet Case Study data that we used in previous exercises. Select soundings from Davenport, Iowa (74455) for 1998-06-29 12:00:00Z and 1998-06-30 00:00:00Z. Compare the soundings to the meteorological situation shown by the radar.
2. Create a plan view display of 500 mb parameters for 00Z today. Overlay the contours of 500 mb geopotential heights from the initial time step of the 00Z NAM grid.
3. Load in the latest 12Z RUC model data. Create a `Grid Skew-T` display. Load in several 12Z soundings scattered over the model domain and create Skew-T displays of the data. Compare the model soundings with the observed soundings.

3.10 Trajectory Data

The IDV supports a variety of aircraft track data and displays.

3.10.0 Trajectory Display

The IDV can display a variety of flavors of trajectory data - aircraft tracks, ships, buoys, etc.

1. If other displays and data are already loaded, select the **Edit** **Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load some aircraft track data.
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `Files` tab.
   - Go to the point data directory: `/data/idv/trajectory/`.
   - Select `track.nc`
   - Under the `Data Source Type` pulldown menu, select "Track files"
   - Press `Add Source`
3. Create the display
   - In the `Fields` panel open the `Track` node and select `Dew/Frost Point Temperature`.
   - In the `Displays` select `Track Colored By Parameter` and press `Create Display`
4. Add a background image to see where this track is with the **Displays** **Maps and Backgrounds** **Add Background Image** menu item.
5. Leave the data source and display as is.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.10.1 Trajectory Point Display

Trajectory data can be displayed as individual point observations.

1. If you do not have track data loaded see Trajectory Display.
2. For performance reasons we want to subset the point data.
    - Right click on the `track.nc` entry in the `Data Sources` section.
    - Select **Subset Point Data**
    - Select all fields under `Current Fields` by clicking in the list and pressing Control-A.
    - Remove all selected fields by pressing `Remove`
    - Select the fields of interest. e.g., `Altitude` down to `Dew/Frost Point Temperature` with successive Control-clicks or Shift-Clicks.
    - Press `Add`
    - Press `OK`
3. Create the `Point Data Plot` display.
    - In the `Fields` panel select the `Point Data` field.
    - In the `Displays` select `Point Data Plot` and press `Create Display`
    - When the IDV asks if you want to see all of the times select `No`.
    - Change the `Station Model` used to be `Airplane`
4. Change the track time display
    - Bring up the display control window for the `Track` display.
    - Press Time Mode `Change` button.
    - For End Time choose `From Animation Time`
    - For Start Time choose `Relative to End Time`
    - Change the Offset for the start time to be -30 minutes.
5. Turn on animation.
6. For the point display change the `Station Model` used to be `TrackDewpoint`

3.10.2 Storm Tracks

The IDV can display tropical storm track data from the ATCF FTP database.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load in the ATCF data and create the Storm Track display.
   - Normally selecting **Data Special ATCF Data** will load in a data source with a predefined path to the National Hurricane Center ftp site.
   - But, for stability and speed in the workshop we copied over the data for 2008 and 2007 to our ftp site so...
   - In the URL chooser type in: *ftp://ftp.unidata.ucar.edu/pub/idv/data/atcf/*
   - Under `Data Sources of Type` select "ATCF Tropical Storm Data" (hint: click in the pull-down box and enter an 'a').
   - Load in the data source.
   - Create a `Storm Track` display.
3. Choose **2008 Basin:AL IKE** and select `Load Tracks`
4. Note the huge number of forecasts. We want to pare this list down.
   - Bring up the Properties dialog for the display. (**Edit Properties** menu).
   - In the "Techs to use" tab select all of the Techs in the "Use" list. Click and press control-a.
   - Press `Remove`.
   - Select a number of Techs in the "Don't use" list and press `Add`. Make sure you include GFDL and HWRF in your selection.
   - Apply these selections by pressing `OK`.
5. Turn on the `Forecasts` checkbox to show the forecast tracks.
6. Note, the forecasts tracks are showing forecast hours which results in a complicated display. Turn this off by selecting a "None" for the Points Layout model.
7. Turn off all but one Forecast and set the `Animation Mode` to "Off".
8. Let's look at charting the forecasts.
   - In the `Storm Chart` tab select GFDL and HWRF and Observation in the Techs list.
   - Select one of the later times, e.g., 2008-09-10.
   - Select both `Min Pressure` and `Max Windspeed`.
   - Turn on `Use Difference`.

3.11 Miscellaneous Displays

3.11.0 Drawing Control

The IDV provides a 3D drawing tool.

1. First, start fresh and go to a `CONUS` projection.
2. Create a `Drawing Control` with either the **Displays Special Drawing Control** menu item or with the toolbar button.
3. When the `Drawing Control` first comes up it is in polygon drawing mode. Note the button.
4. Draw some shapes
   - In the main display click and drag to draw.
   - Select the button and check on the "Straight" button to draw a straight line polygon. Click and drag in the main display. Press the "space" key to add a polygon point.
   - Select the button to draw a rectangle. Click and drag in the main display.
   - Go to smooth polygon mode with the button.
   - Tilt the display and draw some more shapes. Note how you are drawing in 3D.
   - Go back to the overhead view (Control-R).
5. Go into move mode by pressing the the button. Move some of the shapes. Note: You have to select on the corners for squares, at the start for text, etc.
6. Select and delete the shapes.
   - Select the button to go into selection mode.
   - Click on one of the shapes to select it. You should see small blue highlight rectangles.
   - Press Control-X to cut the shapes.
   - You can select multiple shapes by pressing the control key when you click on them.
   - You can select all of the shapes by pressing Control-A
   - Delete them by pressing Control-X
7. Change the properties of a shape.
   - Add a new text shape.
   - Select it with the mouse or with a Control-A. Note: to select text you need to click at the beginning of the text.
   - Press Control-P to bring up the properties dialog.
   - Change some of the properties and press `OK`
8. Creating text shapes
   - Select the T button and click in the main display to add text.
   - In the dialog box add in some html. e.g.:

     ```
     <h2>Some example text</h2>
     <a href="http://www.unidata.ucar.edu">Click here</a>
     ```
   - The HTML is shown in the preview window.
   - Press "OK" when you are done.
   - Try clicking on the link.
9. Add a Quicktime movie.
   - Select the movie reel icon:
   - Click in the display. In the File dialog select /data/idv/misc/storm.mov
   - Note: Not all movie encodings can be displayed in the IDV.
10. Drawing in coordinate systems.
    - There are 4 coordinate system you can draw in: Lat/Lon/Alt, Lat/Lon (with fixed Altitude), X/Y/Z, X/Y (with fixed Z).
    - Draw some shapes.

- Change the projection to a world projection (**Projections Predefined World** menu item) and note how the shapes are reprojected.
- Select X/Y from the `Location` menu.
- Now draw some more shapes.
- Change the projection to back to CONUS (**Projections Predefined US CONUS** menu item) and note what shapes get reprojected.

11. Export/Import
    - You can export and import drawings with the **File Export** and **File Import** menu items.
    - Try exporting your drawing to the file `workshop.xgrf`.
    - The xgrf file can be loaded as a data source or imported into an already existing `Drawing Control`. Let's try loading it as a data source.
    - Remove the existing `Drawing Control`
    - In the `Data Source Chooser` go the `Files` tab and select the `workshop.xgrf` file.
    - You should see a new `Drawing Control` created with what you exported.

Footnotes:
To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

3.11.1 Doing More with the Drawing Control

You can draw across time steps.

1. First, start fresh, removing all displays and data
2. For this exercise we created a "Favorite Bundle" that has the Davenport Radar displays. We'll discuss favorite bundles later. Open this with the **Displays Favorite Bundles Toolbar Workshop Davenport Radar** menu item. This should load in the Davenport radar and displays we used earlier.
3. In this exercise we are going to annotate this radar display with the `Drawing Control`.
4. Create a `Drawing Control` with either the **Displays Special Drawing Control** menu item or with the  toolbar button.
5. Setup the control
   - Choose "Lat/Lon" for the Location.
   - Move the Z Position above the Bottom setting.
   - Select the `Draw in current time` checkbox.
   - Under the `Style` tab change the color to something that will stick out, e.g., cyan
   - Change the line width to 6
6. Annotate the radar display
   - Select a cell in the Bow Echo.
   - Circle the cell with a polygon.
   - Step the time animation forward one step. Note: the previous polygon is no longer shown.
   - Circle the cell of interest.
   - Repeat.
7. Now, start the time animation.
8. Stop the time animation and select the `Show all` button on the `Drawing Control`

Footnotes:
To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

3.11.2 Weather Text Products

Weather text products produced by the National Weather Service and other international agencies can be accessed through the IDV.

1. Load in the **Workshop Current U.S. Weather** bundle from the toolbar menu.
2. Zoom in over an area experiencing some weather.
3. Select the **Data Special Weather Text Products (from server)** menu to open the Weather Text Product control.
4. Expand the `Public Products` category and select the `Area Fcst Discussion` product. This will plot some stations in the main view window.
5. Click on one of the stations in the view window to see the latest discussion.
6. Select the `Zone Forecasts` product for the same station.
7. With the Zone Forecast displayed, click on the HTML tag. This presents an web-ified version of the forecast.
8. Select the `Show Glossary` checkbox. Click on one of the glossary links.
9. Expand the `Observed Data` category and select the `Surface Hourlies` product.
10. Select the `Display` tab in the control window and increase the density of the stations shown. Select the `Layout Model` to change the way the stations are shown.
11. Go back to the `Products` tab in the display and select the `Text` tab.
12. Select several stations (use Ctrl-Click) in the main view window
13. Change the `Time Covered` to 6 hours.
14. Spend some time trying other products/times.

3.11.3 Web Map Servers

The IDV provides access to Web Map Servers (WMS).

1. Start fresh by clearing all displays and data and going to a CONUS projection.
2. Use the **Displays Maps and Backgrounds Add Background Image** menu to load in the background image control. The default view is the Blue Marble image from NASA.
3. The WMS display allows you to save off the displayed image in either IDV's ximg format or Google Earth's kml/kmz format.
    - Change the Layer in the WMS display control to Blue Marble Dynamic.
    - From the WMS display's **File Save** menu choose **Save As Image XML/KML File...**.
    - Enter `bluemarble.ximg`.
    - Now, from the Files data source chooser press `Update` and select `bluemarble.ximg`.
    - Create a 3 Color (RGB) Image.
    - Toggle visibility of the displays.
    - Remove this display.
4. Let's look at a Topographic map.
    - Change the Layer in the WMS display control to Topographic Map
    - Note that all we get is a white background. This is because the size of the view is too large and the WMS we are accessing just returns a white image.
    - Zoom in over Colorado .
    - Note that the WMS Control automatically fetches a new image when the view changes. This can be turned off with the `Auto-Reload` button.
    - Keep zooming. What happens? The screen went blank. This is because we just zoomed through the surface of the earth.
    - We can fix this by changing the projection. Choose the **Projections Use Displayed Area** menu item.
5. Let's find out how far it is to walk to the cafeteria in FL2
    - Change the `Layer` to `Urban Area`
    - First we need to find our building. **View** menu on the display control window select **Go to Address**. Note: you can also do this from the **Projections** menu in the main view window.
    - Enter our building's address:

      ```
      3300 Mitchell Lane, Boulder, CO
      ```
    - Press `OK`
    - We should now be looking at FL4. Move the display so FL4 (control key/right mouse drag) is in the lower left.
    - Add a `Range and Bearing` control with the **Displays Special Range and Bearing** menu.
    - Move the endpoints to figure the distance.
    - Lets see how many furlongs it is.
        - From the `Range and Bearing` **Edit** menu select **Change Display Unit...**.
        - Enter "furlongs" and press `OK`.
6. Now, go try to find your home. The `Go to Address` dialog can take addresses, zip codes, city/state names or lat/lon points. Mousing over the `Address` field will show a tooltip with the different formats available.

Footnotes:

To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

3.11.5 Locations

The IDV provides a large location database.

1. Clear all data and displays.
2. First, turn on the states map and change projection to Colorado.
   - Bring up the control window for the `Background Maps`
   - Select "Hi-Res US" and close the window.
   - Change projection to the Colorado projection with the **Projections Predefined US States West Colorado** menu item.
3. We're going to look at Colorado summits.
   - Select `Summits` from the **Displays Locations US State locations C Colorado Land features** menu.
   - The `Location` display control either shows locations with a label/shape or with a station model. Select the `Station Model` button and, if not selected, select the `Location` station model by pressing the `Change` button.
   - Turn off decluttering.
   - Things are really cluttered. Let's use a different station model. Select the `Point` station model.
4. Let's only look at summits greater than 14000 feet.
   - First, lets find out what properties these locations have. Click on a point to see the properties in the `Locations` tab.
   - Select the `Filters` tab.
   - Under property select `elev`
   - Select ">"
   - Enter 14000 for the value.
   - Press `Apply Filters`
   - Now, switch back to the `Location` station model.
5. Let's look at some volcanoes
   - Clear all data and displays and select
   - Select **Displays Locations Earth Volcanos**
   - Float the Locations display control window with the menu **View Undock From Dashboard**.
   - Under the `Display: Predefined:` change `Id:` to "Name"
   - Select the `Locations` tab to see a list of locations.
   - Click on a row to view different volcanos. Shift-click to zoom in. Control-click to zoom out.
   - Add the real-time USGS earthquake list with **Displays Locations Earth Earthquakes**.

3.11.7 DEM Displays

The IDV supports elevation data from models, USGS DEM format and ESRI's ArcInfo ASCII grid format.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Load some DEM data.
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `Files` tab.
   - Go to the point data directory: `/data/idv/gis/`.
   - Select `dem.grd`
   - Under the `Data Source Type` pulldown menu, select "DEM Files"
   - Press `Add Source`
3. Create a display
   - Select `Color-Shaded Plan View` in the `Displays` section of the `Field Selector` and press the `Create Display` button.
4. Where are we?
   - To find out create a Background Image (**Displays Maps and Backgrounds Add Background Image** or with the toolbar button). Select the "Topographic Map" layer.
   - Humm, now what, we can't see the DEM image.
   - The background image is blocking the DEM display. Unfortunately we can change the Z-level of the background but it is already at the lowest level so change the vertical position of the DEM display.
5. Ok, so now we see that this DEM data set is to the west of Denver.
6. Note that the DEM display is flat, it is just a plan view. Doesn't the IDV do 3D???
   - Yes it does.
   - Remove the DEM display and select the Elevation field for the DEM data in the `Field Selector`.
   - Under `3D Surface` in the `Displays` area select `Topography` and create the display.
   - Try changing the `Display Mode` to points. It looks better if you turn off the visibility of the Topographic map.
7. Now, lets try to combine the Topographic map with the DEM data. i.e., lets drape a map image over the DEM.
   - First, a tricky part: we want to somehow capture the background image over the spatial domain of the DEM data set.
     - Zoom in to the DEM image.
     - In the **File Save** menu of the `Background Image` display control select **Save Data in Cache...**.
     - Leave the name as is and press `OK`
     - Now, we have a snapshot of the Topographic map in the `Field Selector`
   - Remove the `Topography` and `Background Image` displays.
   - In the `Field Selector` select `Formulas` and in the `Fields` list, under `Imagery` select `Image over topography`.
   - In the `Displays` list, under `Imagery` select `3 Color (RGB) Image over topography` and press `Create Display`
   - A new `Field Selector` will popup. Select `Elevation` and press `OK`.
   - Another `Field Selector` will be shown for the image field. Select `Topographic map` under `Cached data`.

- ♦ The Topographic map should now be shown draped over the topography.

8. Try changing the vertical scale to 5000-14000 feet (**View** **Viewpoint** **Vertical Scale** menu item).
9. Lets just highlight the areas that are above 4000 meters.
   - ♦ Add a `Location Indicator` display with the **Displays** **Special** **Location Indicator** menu item.
   - ♦ In the `Altitude` field for the origin enter 4000 and hit return.
   - ♦ Under the `Display` tab turn off the `Visibility` for the X, Y and Z axis. Turn on the `Solid` checkbox for X. Set the `Transparency` to 20%.
   - ♦ What are the names of these peaks?

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.11.8 Doing more with GIS Data

1. Create a three pane view. Show the `Topography Background Image` in one pane. Load in the DEM data we viewed earlier and create a `Color Shaded Plan View` in the second pane. Create a `3D Surface/Topography` display in the third pane.

   Add `Data Transect` displays for the DEM data in each of the 3 view windows. Turn on sharing in each of the 3 `Data Transect` displays with the **Edit Sharing Sharing On** menu item. Move the transects around in the display.
2. Load in a topographic map background and the roads and stream shapefiles for your own county. Look around to see how they match up.
3. Look at hot springs in Colorado. Turn on county maps in the `Background Maps` and load in the location file: **Displays Locations State Locations C Colorado Water Features Spring**

3.11.9 Using Images in the IDV

The IDV can display non-geolocated images and geolocate them.

1. First, start fresh
2. Load an image.
   - Open the `Data Source Chooser`.
   - In the `Data Source Chooser` click on the `URLs` tab.
   - Enter the URL:

     `http://www.unidata.ucar.edu/software/idv/data/image.gif`
   - Press `Add Source`
   - The IDV will take some time to load this because it defaults to the VisAD data source.
3. Let's look at this image.
   - You should have the image loaded in the `Field Selector` Note the funny field name. This is the VisAD *math type* of the data.
   - Create an `Omni Control` display to see the image.
   - Click on the `Mappings` button.
   - Let's play around with some of these mappings.
4. Now we're going to look at the moon.
   - In the `Data Source Chooser` click on the `URLs` tab and enter the URL:

     `http://www.unidata.ucar.edu/software/idv/data/moon.gif`
   - Let's geolocate the image.
     - In the `Data Sources` in the `Field Selector` window select `Formulas`
     - Under `Fields` open the `System` tab. Select `Geolocate an Image`.
     - Under `Displays` select `3 Color (RGB) Image` and create the display.
     - In the popup enter: 90,-180,-90,180 and press `OK`
     - Now, select the funny `(ImageElement,ImageLine)...` field.
   - Now, a flat moon picture isn't very exciting. Let's create a Globe display window with the **File** **New** **View Window** **Globe Display** **One Pane** menu item.
   - Create the moon image as described above.

Footnotes:
To start fresh choose the **Edit** **Remove All Displays and Data** menu item or the scissors icon in the toolbar.

Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data** **Choose Data** menu.

3.12 NOAA Profiler Network Winds

The IDV can display winds from the NOAA Profiler Network in time-height plots at a station, and as multi-station plots wind barbs at single heights or at multiple heights in the IDV 3D display.

3.12.0 Loading NOAA Profiler Network Data

NOAA Profiler Network data can be accessed in the IDV from ADDE servers. You can load and display several stations and times at once.

1. Setup for Profiler displays
    - If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out..
    - Uncheck the **Projections Auto-set Projection** menu option.
    - Set the projection to Kansas (**Projections Predefined Kansas** menu).
2. Open the Data Source Chooser for NOAA Profiler Network Data
    - Open the `Data Source Chooser`.
    - In the `Data Source Chooser` click on the `Profiler` tab to see the pane for access to NOAA National Profiler Network data on ADDE servers.
    - The map shows Profiler station names. Zoom in to see all stations in Oklahoma and Kansas.
    - Select the four stations Hillsboro, Haviland, Neodesha, and Lamont, one at a time, by holding the Ctrl key down while clicking on the station name. You can put one or more stations in any IDV Profiler display. More about selecting and deselecting stations.
    - In the `Server` box, use the pull down list to select a remote ADDE server with Profiler data. Select adde.ucar.edu.
    - Select `Hourly` in the `Data Interval` box. You can choose from Hourly, 30 minute, 12 minutes, or 6 minute. This sets the time intervals in the display.
    - Click on `Connect`. A list of times will appear; the most recent 12 times are pre-selected.
    - Select the most recent 6 times available. (Shift-click on the sixth time up from the bottom.)
    - Click on `Add Source` to connect the IDV to the data for this Profiler data request.

    The label `Profiler Hourly - 4 stations` appears in the `Data sources` panel in the `Field Selector`, and `Profiler winds` appears in the `Fields` panel.

Footnotes:
Open the `Data Source Chooser` by either:

- Select the `Data Chooser` tab in the `Dashboard`
- Choose one of the menus under the **Data Choose Data** menu.

More about selecting and deselecting stations.

You can de-select stations with Ctrl-click too. You can select a group of stations by a rubber band box. Hold the Shift key down, then left click and drag the mouse. This only selects stations shown on the map; make sure decluttering is turned off to get all stations in a region.

3.12.1 Profiler Time-Height Displays

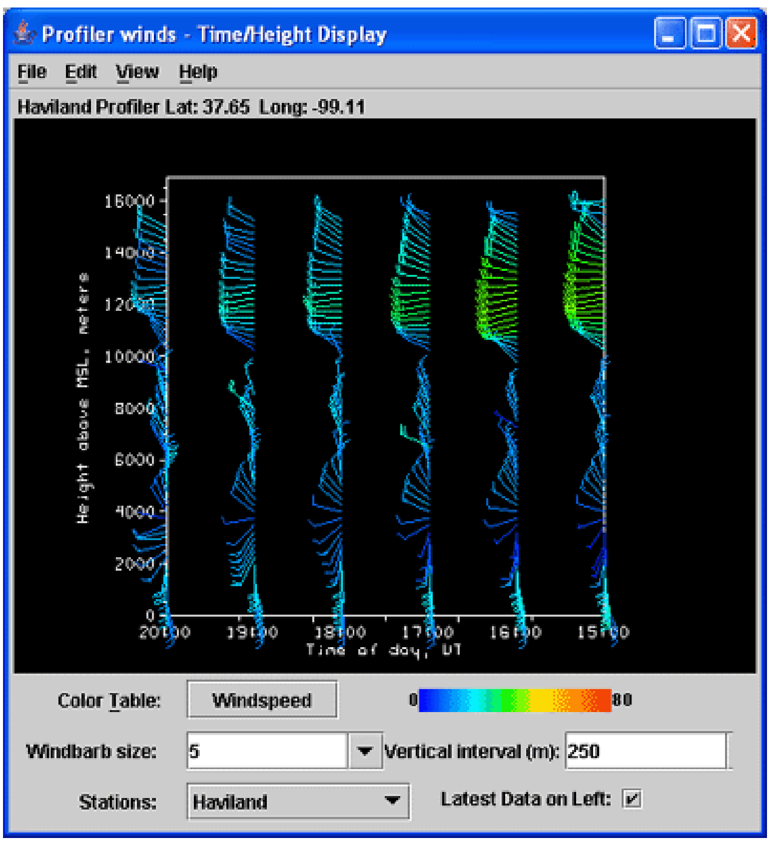
The Profiler Time/Height display shows profiler winds at all levels above one station, for one or more times. This is the conventional display of profiler data.

Winds are shown with the meteorological wind barb symbol. The location of the barb on the plot shows wind observation time, and height above mean sea level; the barb's orientation gives wind direction in a map view sense (up is north; right is east).

If necessary, select Profiler data and times as described in Loading Profiler Data.

1. In the `Field Selector` window create a `Time/Height Display` using the `Profiler winds` field.

   The Profiler Time-Height display and control appears in a new window.



2. Working with the Profiler Time-Height Display

   - Enlarge the display by dragging the edges of the window border with the mouse.
   - Use the `Windbarb size` box to enlarge the wind barbs size to 7.
   - Use the `Vertical interval` entry box to change the vertical separation between wind barbs to 500 meters.
   - Click on the `Color Table` button to get a menu of named color tables supplied in the IDV. Change color table to **Basic->Bright 38**.

- The `Stations` box appears with a list of all stations selected. Use it to switch between stations displayed. View all the stations' plots.
- The checkbox `Latest data on left` is used to change the ordering of times displayed on the x axis.

3. Remove All Displays
    - If other displays are in the main IDV window, select the **Edit** **Remove All Displays** menu item or the icon in the toolbar to clear them out.

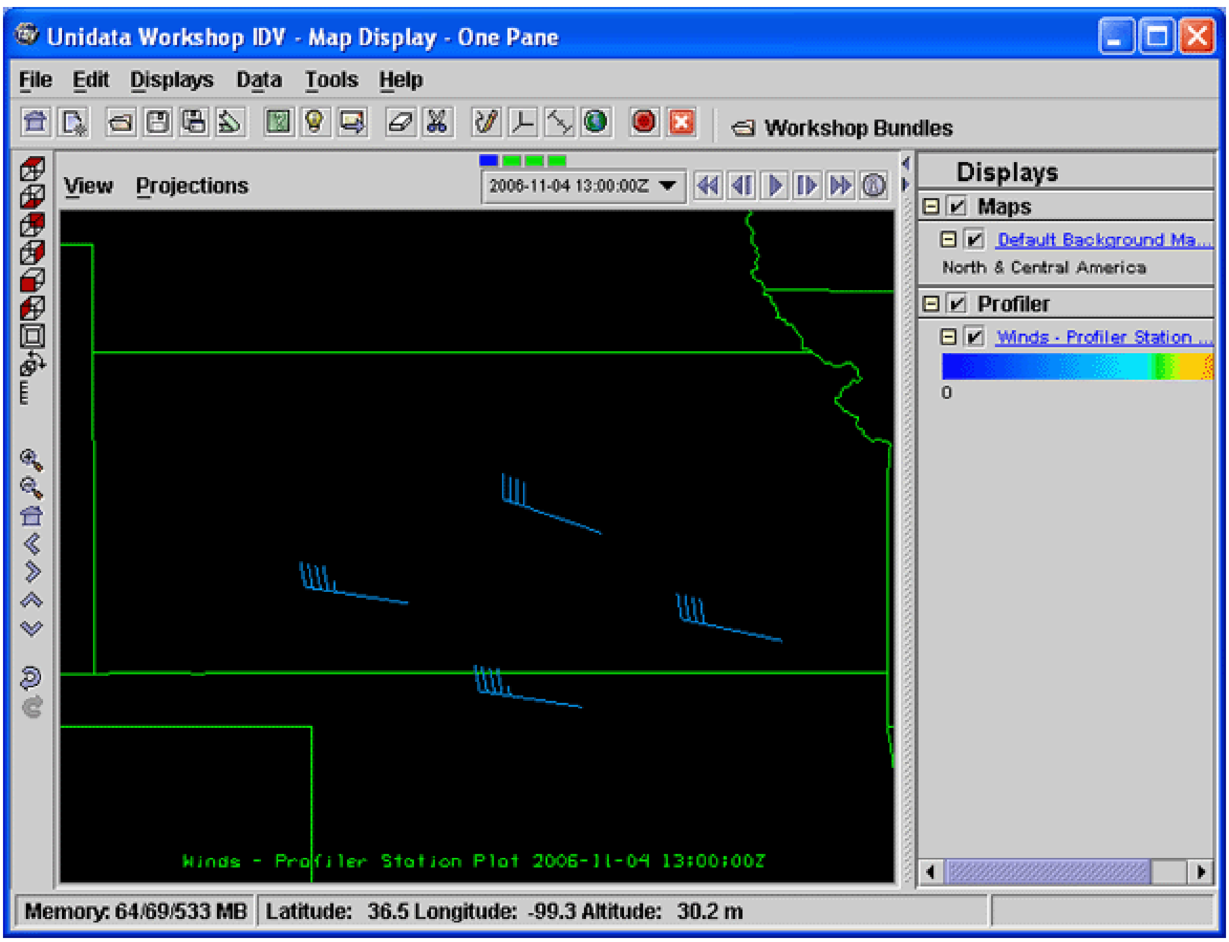
3.12.2 Profiler Station Plot Displays

The Profiler Station Plot shows winds at one level in the atmosphere, above one or more stations.

Winds are shown with the meteorological wind barb symbol. This is a 3D display and the data appears in the correct elevation in the upper atmosphere.

If necessary, select Profiler data and times as described in Loading Profiler Data.

1. In the `Field Selector` window create a `Profiler Station Plot` display using the `Profiler winds` field.



2. Working with the Profiler Station Plot
   - Use the control's **Plan level (m MSL)** box to change the level to 12000 meters.
   - Use the control's `Windbarb size` box to enlarge the size to a size you like.
   - Rotate, and zoom out or in, to see the wind barbs at altitude in the 3D display. Set the viewpoint so you have a good view of the upper level winds from all four stations.
   - Start time animation, then stop animation.
3. Remove All Displays
   - If other displays are in the main IDV window, select the **Edit** **Remove All Displays** menu item or the icon in the toolbar to clear them out.

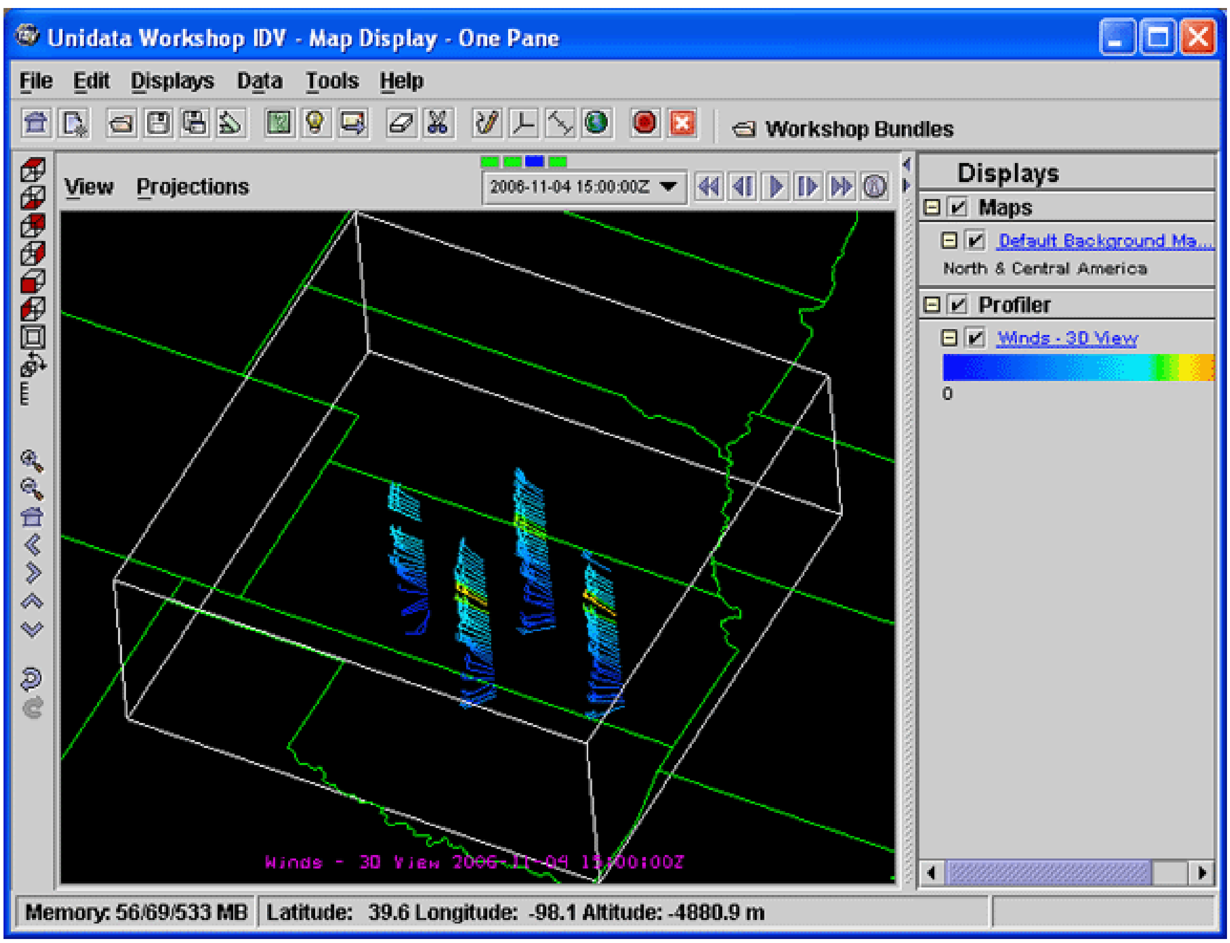
3.12.3 Profiler Three-D View

The Profiler Three-D View shows winds at all levels above one or more stations, in the main 3D view window.

If necessary, select Profiler data and times as described in Loading Profiler Data.

1. If other displays are in the main IDV window, select the **Edit Remove All Displays** menu item or the icon in the toolbar to clear them out.
2. In the `Field Selector` window create a `3D View` display using the `Profiler winds` field.

   The Profiler 3D View appears in the main 3D window. The control window for this display also is created and appears.

3. Working with the Profiler Three-D View

   - Zoom out, rotate, and pan as needed so you have a good view of all the profiler winds at all four stations.
   - Use the control's `Windbarb size` box to set the size to 5.
   - Use the control's `Vertical interval` box to change the vertical separation to 1000 meters.

- Start and stop time animation.

4. Remove All Displays

    - If other displays are in the main IDV window, select the **Edit Remove All Displays** menu item or the icon in the toolbar to clear them out.

3.12.4 Doing More with Profiler Displays

How to merge Profiler winds with other upper air data in one display.

Choices of numbers of stations, background maps, map projection, wind barb size, and vertical interval can make Profiler displays which range from very sparse to very complex. Work with just the data you want to see and try different choices to bring out the best of the Profiler data.

1. Use Data Projection
    - Click on the **View Use Data Projection** menu item in the control window.
    - The display is remapped to show the data in its native data projection. This is very useful for Profiler plots if your previous projection was not suitable for showing the data. For example, if you were looking at model output over North America before you looked at Profiler, and the menu item **Projections Auto-set Projection** was not checked on, the Profiler display is very small.
2. Merging Data
    - Make a 3D View of Profiler winds from a few stations, in its own projection.
    - Click off the menu item **Projections Auto-set Projection**, so other data put in this display does not alter the projection.
    - Select a data source of model output, for the same time and area.
    - Make a display from the model analysis of wind barbs at upper air levels.
    - Use time animation, and viewpoint rotation, to compare the wind data from two sources.

3.13 Working with WRF Output

WRF output can be analyzed and visualized in a number of ways, both interactively and from batch script processing. The first part of this section will focus on using the IDV to interactivly display and analyze WRF output; the second part will focus on the use of the Jython scripting language.

3.13.0 Interactive

3.13.0.0 Loading WRF Output

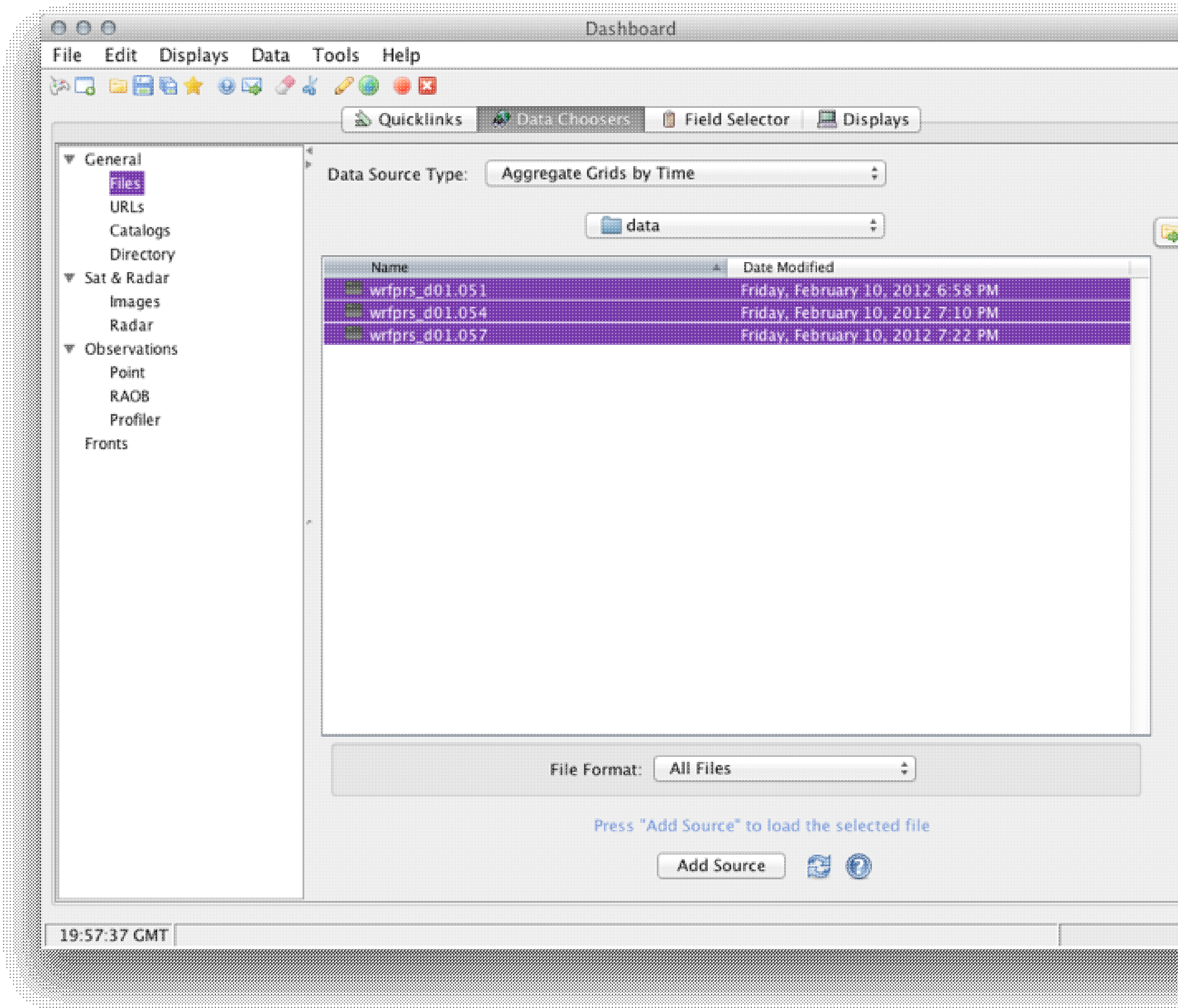
Getting your WRF post-processed data into the IDV environment.
If you have participated in the WRF tutorial, then you are likely familiar with the output from the Hurricane Gustav simulation, which is what we will use in this tutorial. These data are on the thumb drive you received as part of the workshop, although any WRF post-processed output should do fine. While the IDV can handel the staggered grids produced by WRF, it's **highly** recommended that you:

- post process the raw output to get your data on a non-staggered grid.
- use netCDF or grib output.

While the IDV can handle staggered grids, it's a large performance hit to let the IDV de-stagger themfor you on the fly when performing calculations.

The first step in using the IDV for visualization and analysis is to get your data into the the IDV environment, which is the focus of the first section of this tutorial.

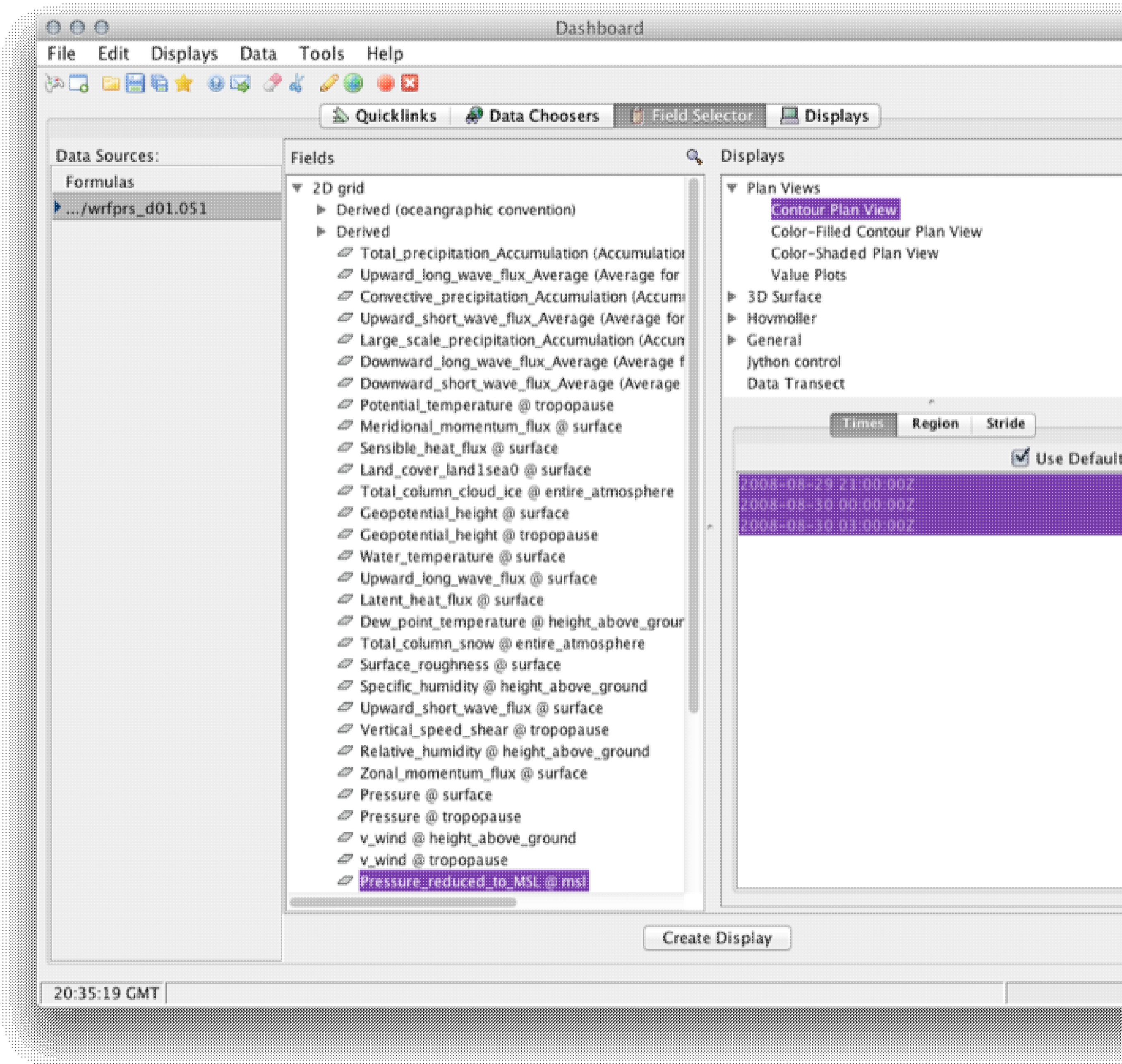
The IDV GUI has two main windows: the Dashboard and the View Window. The View Window is where you will see and manipulate displays you create. The Dashboard is where you will be able to load your data and easily access all controls for manipulating the properties of variables you wish to display (like setting contour properties, adding color scales, changing color tables, etc). In order to get your WRF output into the IDV from the GUI interface, you will need to start the IDV and go to the Dashboard.



The display above shows the Dashboard window of the IDV. The `Data Choosers` tab has been selected, and we are going to directly load files off of our computer (note `Files` under `General` is selected on the left hand side of the Dashboard). Follow these steps to begin the process of loading your data into the IDV.

1. Locate your WRF output in the file browser and select the files you wish to load (hold shift to select multiple files).
2. Make sure that `Data Source Type` is set to `Aggregate Grids by Time`.
3. Click `Add Source`.

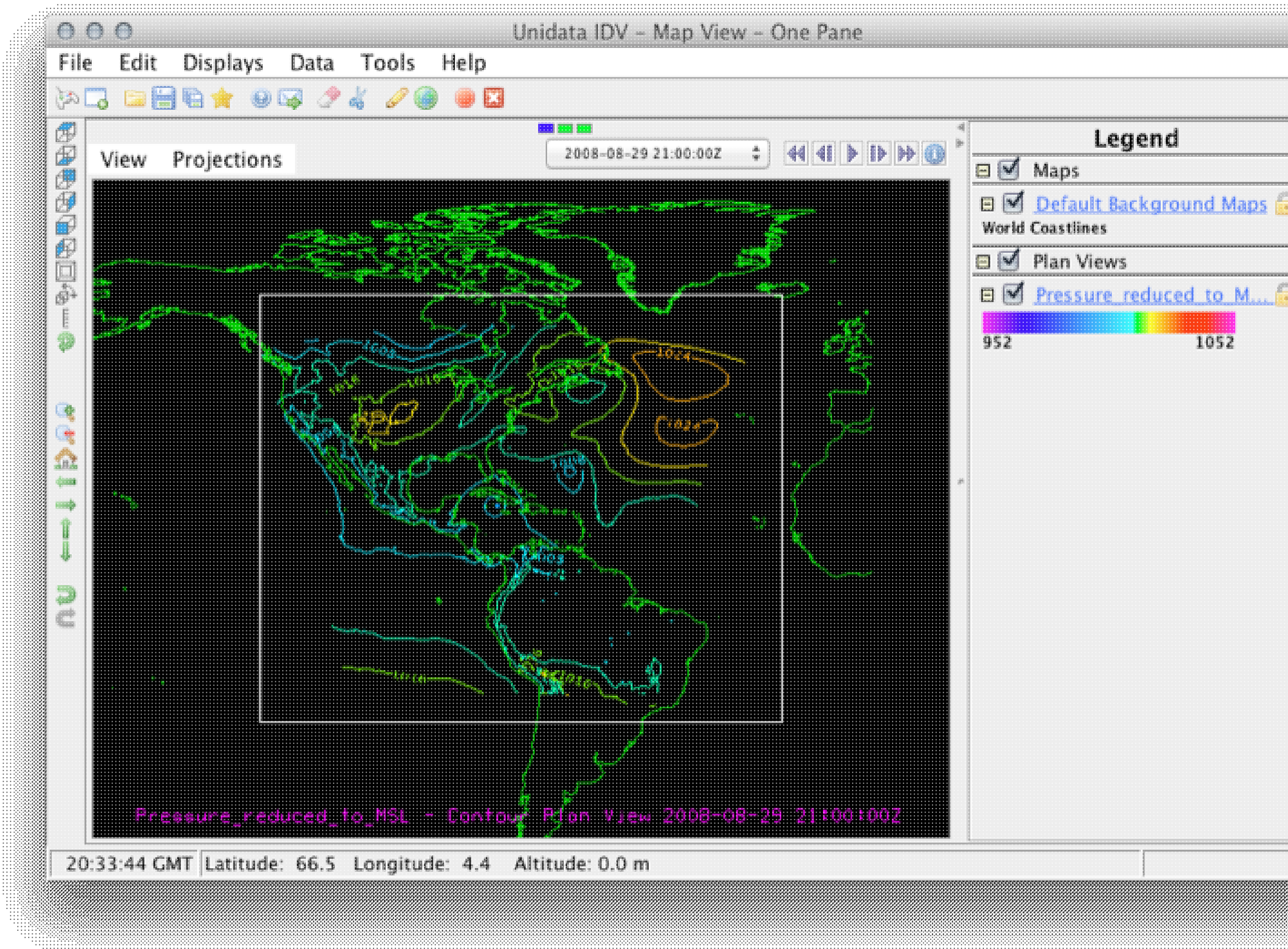
Once you click `Add Source`, the IDV will switch you from the `Data Choosers` tab to the Field Selector tab.



The `Field Selector` tab is where you will select the fields to analyze and types of images to generate. In the screenshot above, the 2D field `Pressure_reduced_to_MSL @ msl` (list on left side of the Field Selector tab) has been selected, and a display type **Plan Views Contour Plan View** (list on top right side of the `Field Selector` tab) has been chosen. For now, we will leave spatial and temporal subsetting as default (i.e. no subsetting) and load the entire grid. For more information on subsetting, see the Field Selector page. The final step is to push the `Create Display` button.

Note: up until this point, very little information has been loaded off of your disk into memory. Once you press the `Create Display` button, data for the field you selected will be read off of disk - depending on the size of your data set, this may take quite some time!

Once the data has been loaded, the IDV will generate an image in the View Window. Congratulations, you've just loaded your WRF data into the IDV! Next, let's explore how to save images and generate animations using the IDV.



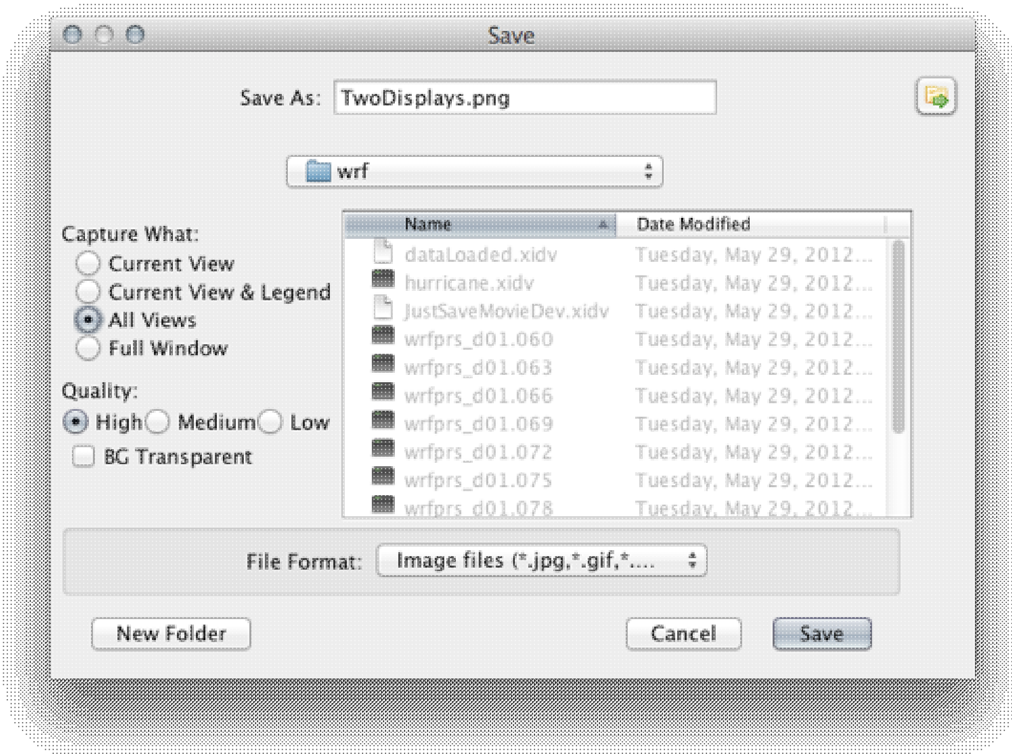
3.13.0.1 Saving WRF Displays

Saving the displays created in the IDV.
In the previous section, we loaded some WRF output and created a very simple contour display. In this section, we will explore how to save images and animations of our displays. However, before proceeding, let's create a `multipanel display` which extends our previous display of `Pressure_reduced_to_MSL @ msl`.

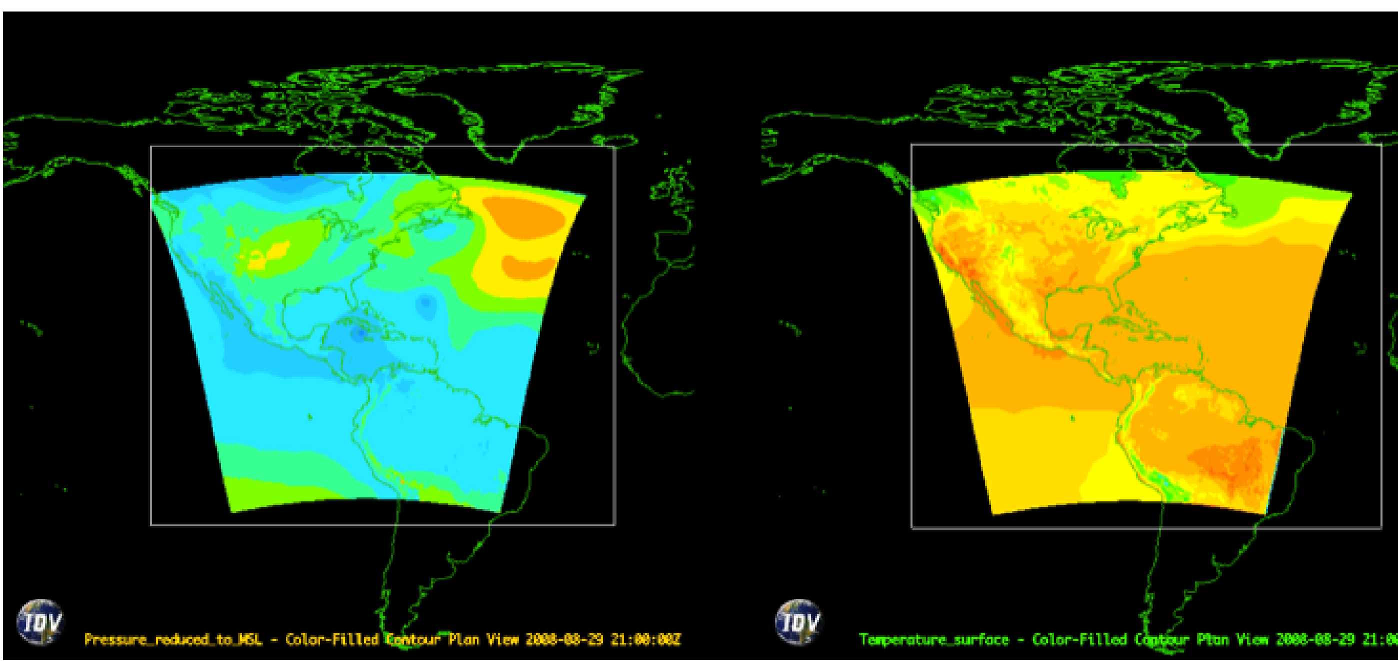
1. From the `Dashboard`, click **File New View Window Map View Two Panes**.
2. Go to the `View Window` and click on the left display (look for border around the window to turn blue).
3. Go back to the `Field Selector` in the Dashboard and create a `Color-filled Contour Plan View` of `Pressure_reduced_to_MSL @ msl`.
4. Go to the `View Window` and click on the right display (again, look for border around the window to turn blue).
5. Go back to the `Field Selector` in the Dashboard and create a `Color-filled Contour Plan View` of `Temperature @ Surface`.

Now, let's save both images:

1. From the `View Window`, click **View Capture Image**. Note that since we are going to capture both images, it does not matter from which display we click `View`. If you wish to capture a single display, make sure to use the `View` menu from that display.
2. An image save dialog box will appear. Give the image a name (TwoDisplays.png), select `Capture What : All Views`, and set the quality to `High`.

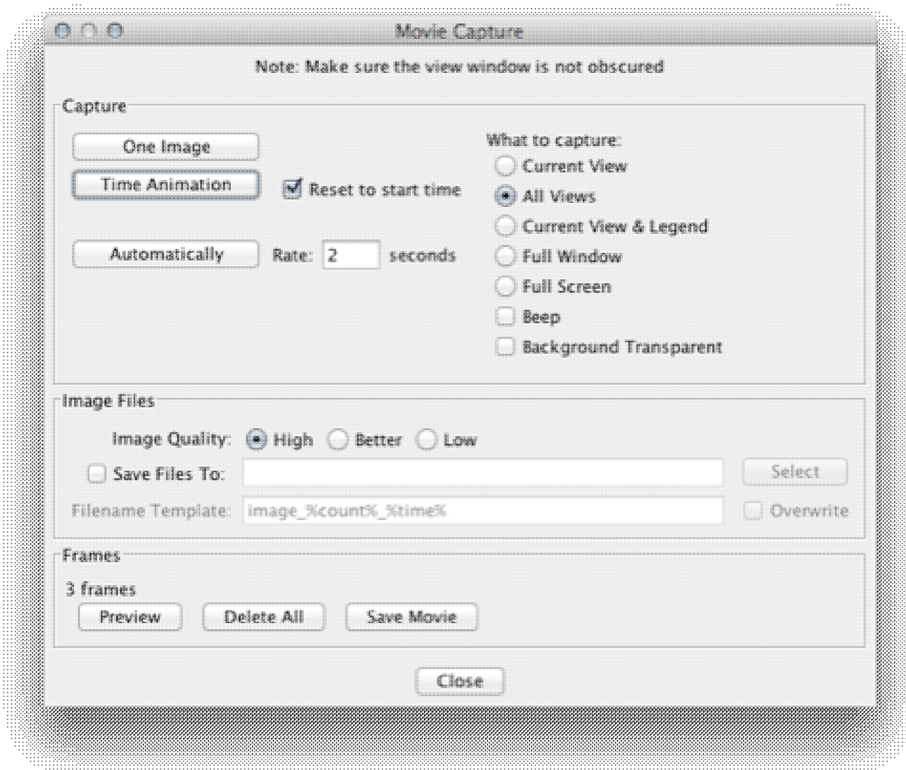


3. Click `Save`. Note that the size of the image will match the size (pixel dimensions) of the window on your display.

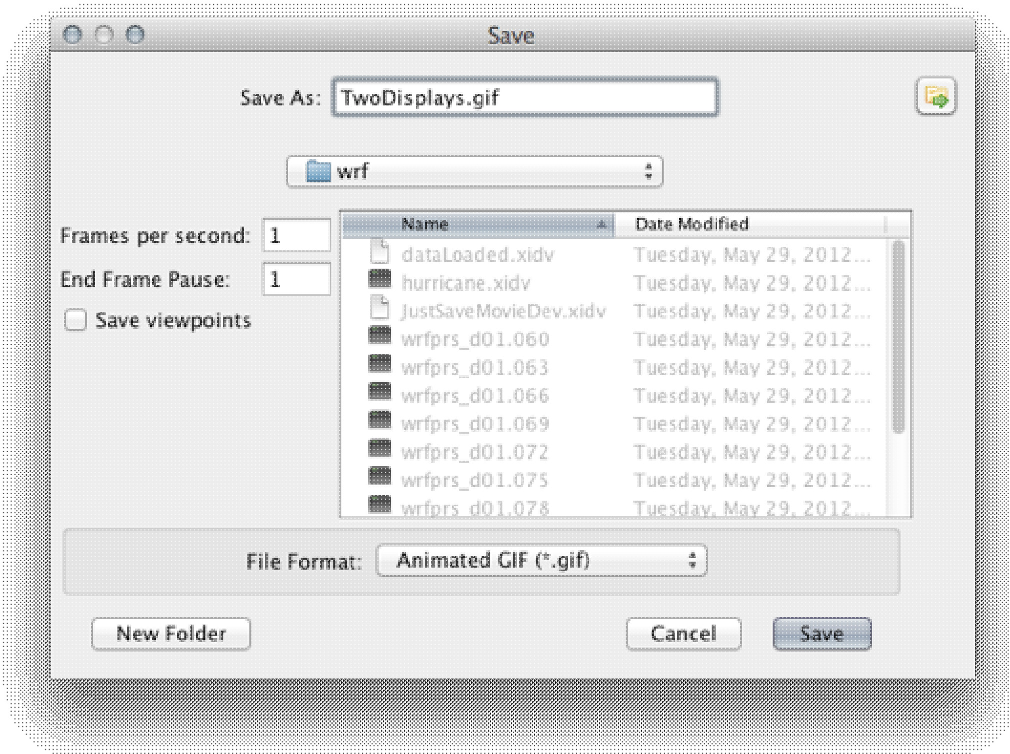


A similar procedure is used to capture an animation of your displays:

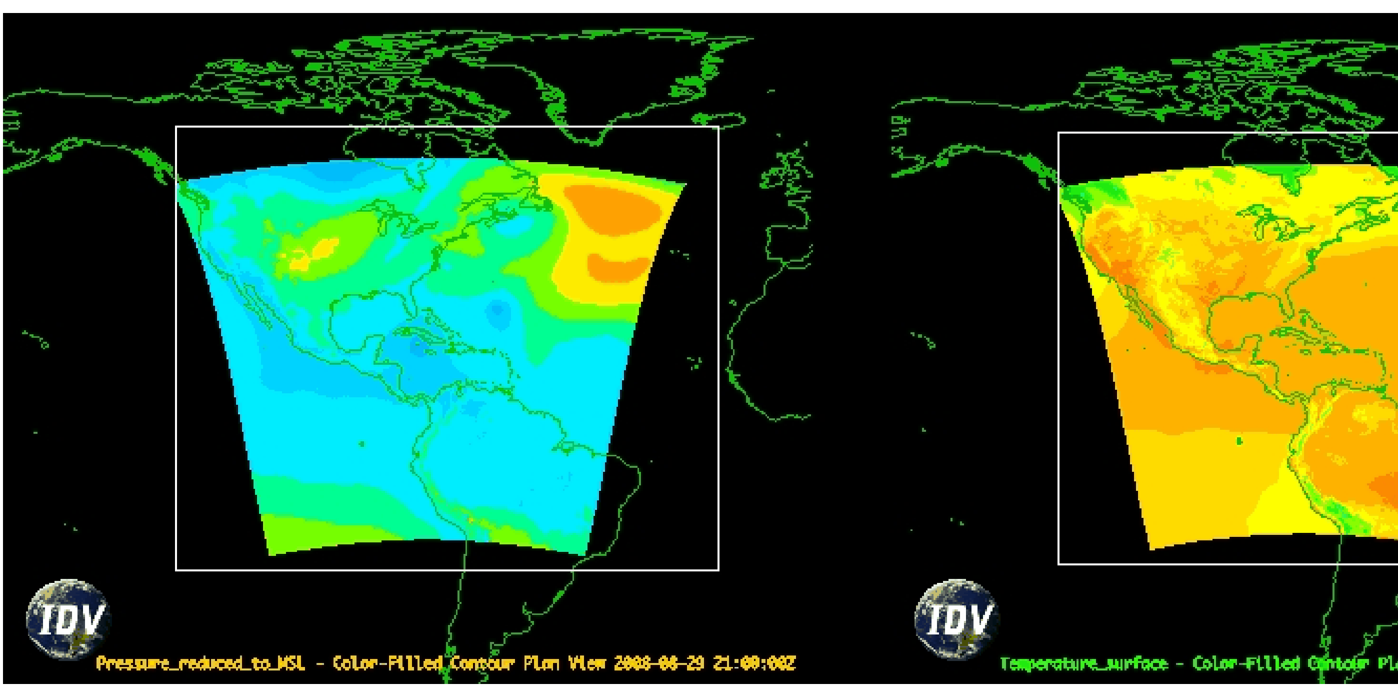
1. From the `View Window`, click **View** **Capture** **Movie**. This will cause the `Movie Capture` dialog box to appear.



2. Make sure `What to capture` is set to `All Views`, and click the `Time Animation` button. The view window will come to the front of you desktop display and animate. Then, a Save dialog box will appear.

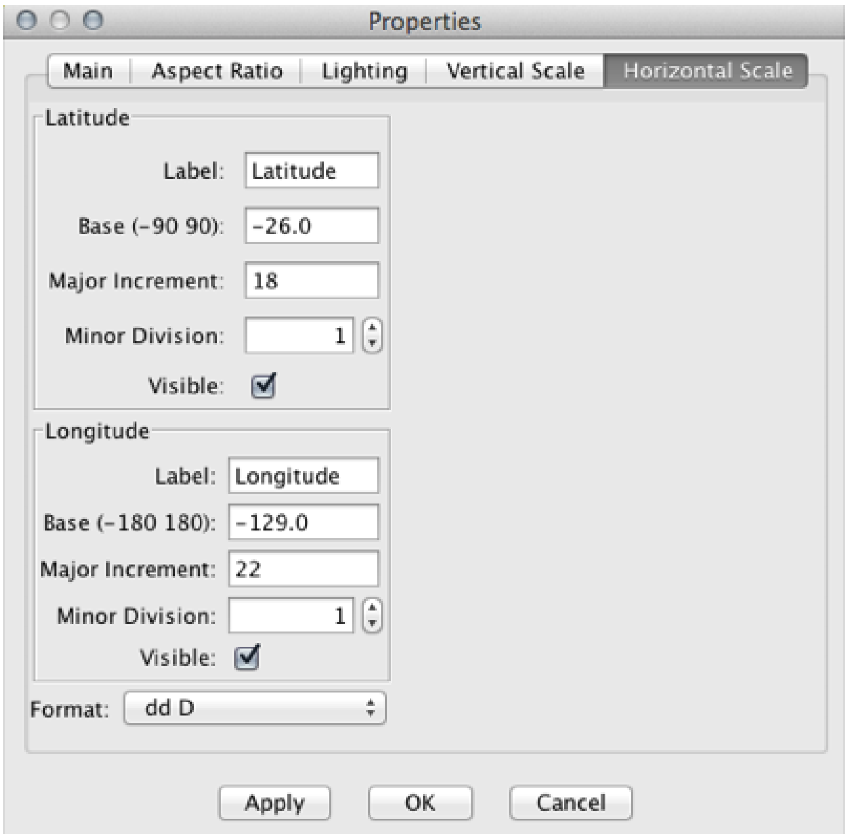


3. Set the `Frames per second` to 1, and `End Frame Pause` to 1.
4. Select the file format (.gif, .mov, .avi, etc.) and name your file.
5. Click `Save`.

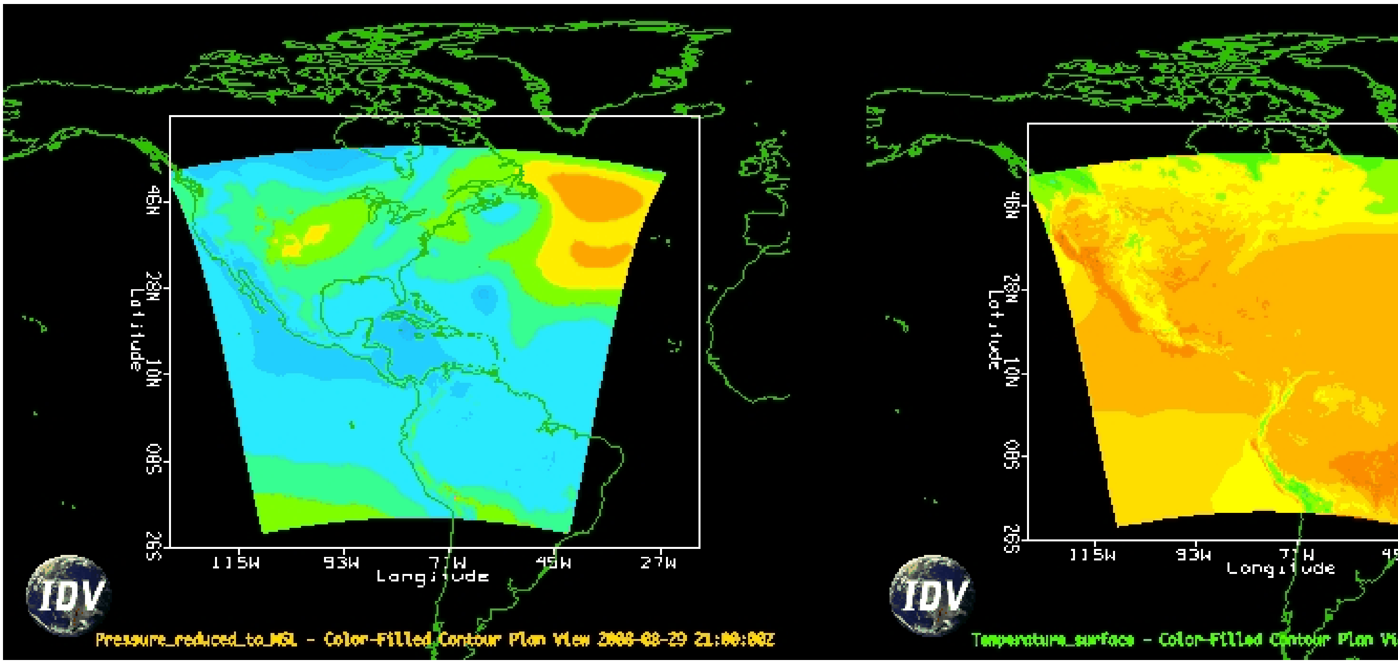


To add axis labels to your images. To display the axis labels:

1. Select the display to which you wish to add labels and click **View Properties**.
2. Click on the `Horizontal Scale` tab and turn on the `Latitude` and `Longitude` scales by clicking the `Visible` box under each section, and then click `Apply`.



3. Finally, save your display (either image or movie), and breathe a sigh of relief.


46N
28N
10N
08S
26S
Latitude
115W
93W
71W
49W
27W
Longitude
IDV
Pressure_reduced_to_MSL - Color-Filled Contour Plan View 2008-08-29 21:00:00Z
46N
28N
10N
08S
26S
Latitude
115W
93W
71W
Longitude
IDV
Temperature_surface - Color-Filled Contour Plan Vi

3.13.0.2 WRF Grid Displays

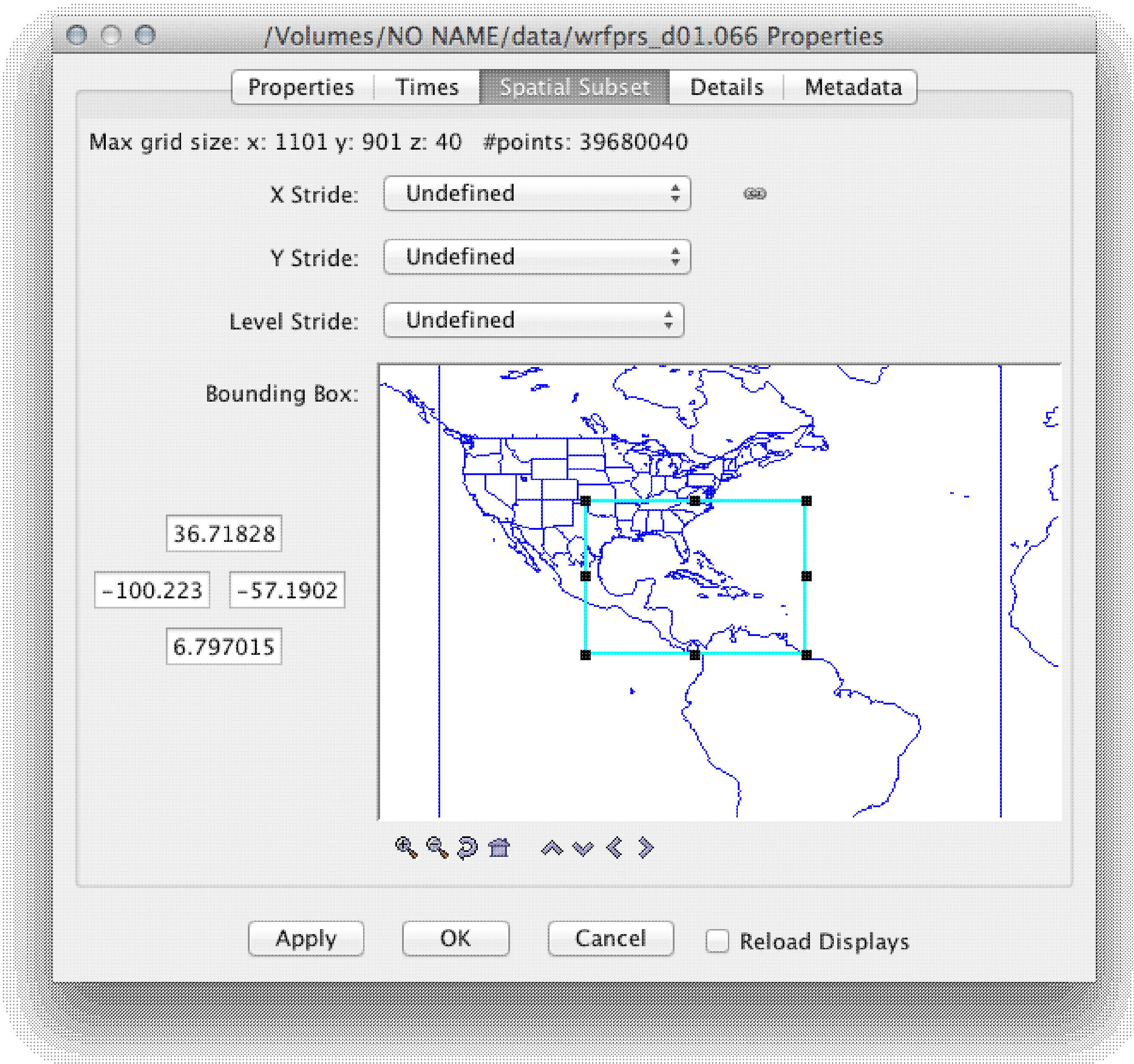
Displays for WRF post-processed data.
The following are examples of some of the different kinds of displays available in the IDV. For more information how to manipulate the display, simply click the image to see the User Guide section for that particular display type. The general format of this section will be an image, followed by the steps needed to reproduce the image. However, before diving into creating different displays in the IDV, let's take a moment to discuss how to handle large data sets.

Preparation: Dealing with Large Data
As you may have experienced, it can take quite some time to get the data into the IDV and render the displays. One way to speed this up is to spatially subset the data over an area of interest. For our case, the Gulf of Mexico is where the 'action' is, so let's subest for that particular region:

1. Clear any existing data sources and displays by clicking the scissors symbol found just below the Main Menu bar on either the Dashboard to Main View window.

Reload your WRF output using the following steps:

2. From the `Data Choosers` tab in the Dashboard, locate your WRF output in the file browser and select the files you wish to load (hold shift to select multiple files)
3. Make sure that `Data Source Type` is set to `Aggregate Grids by Time`
4. Click `Add Source`
5. In the `Field Selector` panel, **right-click** your WRF output under `Data Sources` and click `Properties`.
6. From the Properties box, select the `Spatial Subset` tab and use your mouse to click-and-drag a rectangular box on the `Bounding Box` map display.
7. After selecting your spatial subset region, click `OK`.

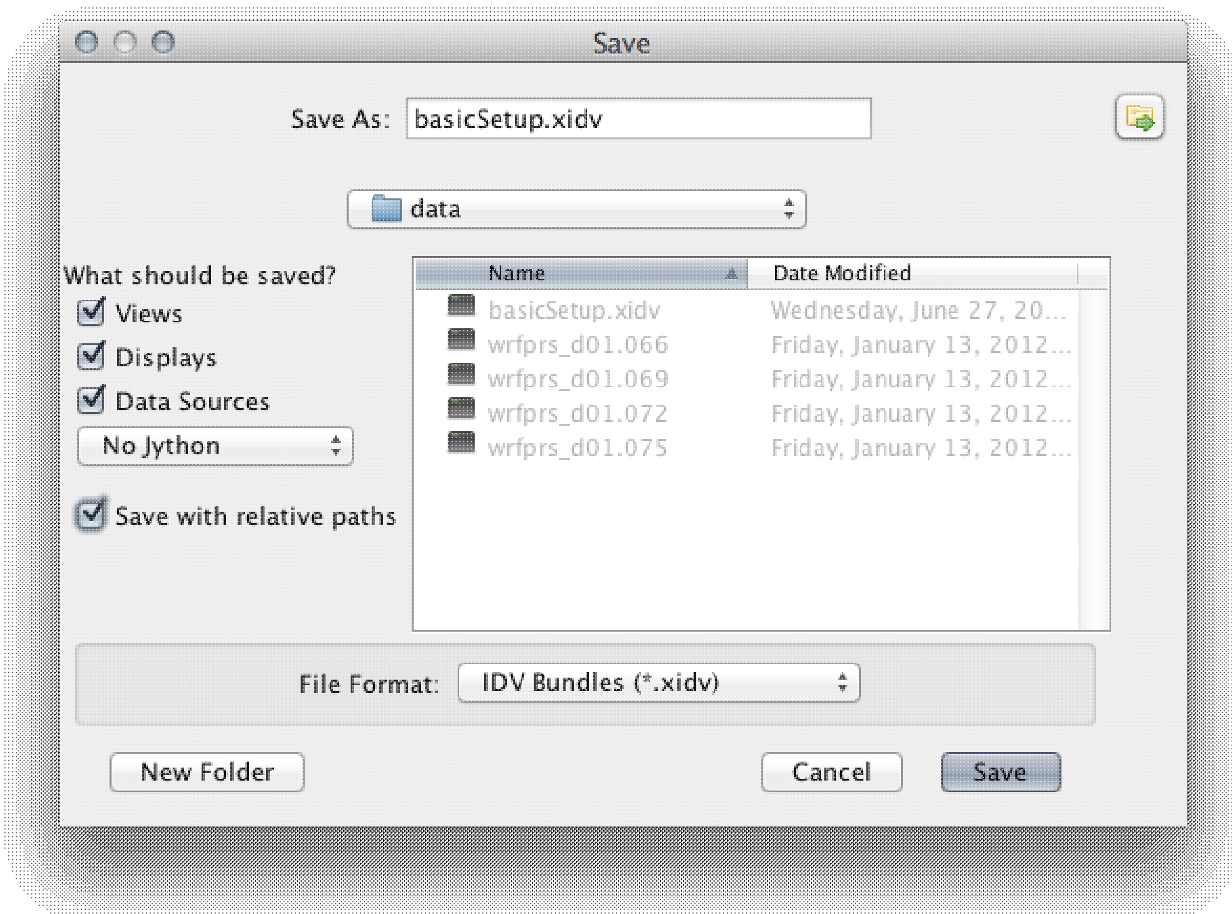


Note: Even with spatial subsetting in place, some displays (particularly the three dimensional ones) will take quite some time to render — consider using only one time step when producing these views.

Many of the example images used below take advantage of the NASA blue marble background image. If you wish to use blue marble, simply click **Displays** **Maps and Bounds** **Add Background Image** and select `Blue Marble`. A second way to add the blue marble background is to click the globe icon found just below the Main Menu Bar. Both of these methods can be used from either the Dashboard or the Main Window.

It may seem like we've done a lot of work to setup our environment (loading data, subsetting, adding a background, etc.). To aid in quickly restoring our environment, we will use the bundle feature of the IDV. A bundle is a way of saving the state of the IDV (current data sources, loaded maps, etc.). You can create a bundle that points to local or remote data (an .xidv), or you can create a bundle that actually contains the data (a .zidv), the latter of which is useful when sharing displays of data you've generated yourself, as it ensures your colleagues have the data needed to view the display. For our purposes, let's create a bundle that points to the WRF output on our memory stick:

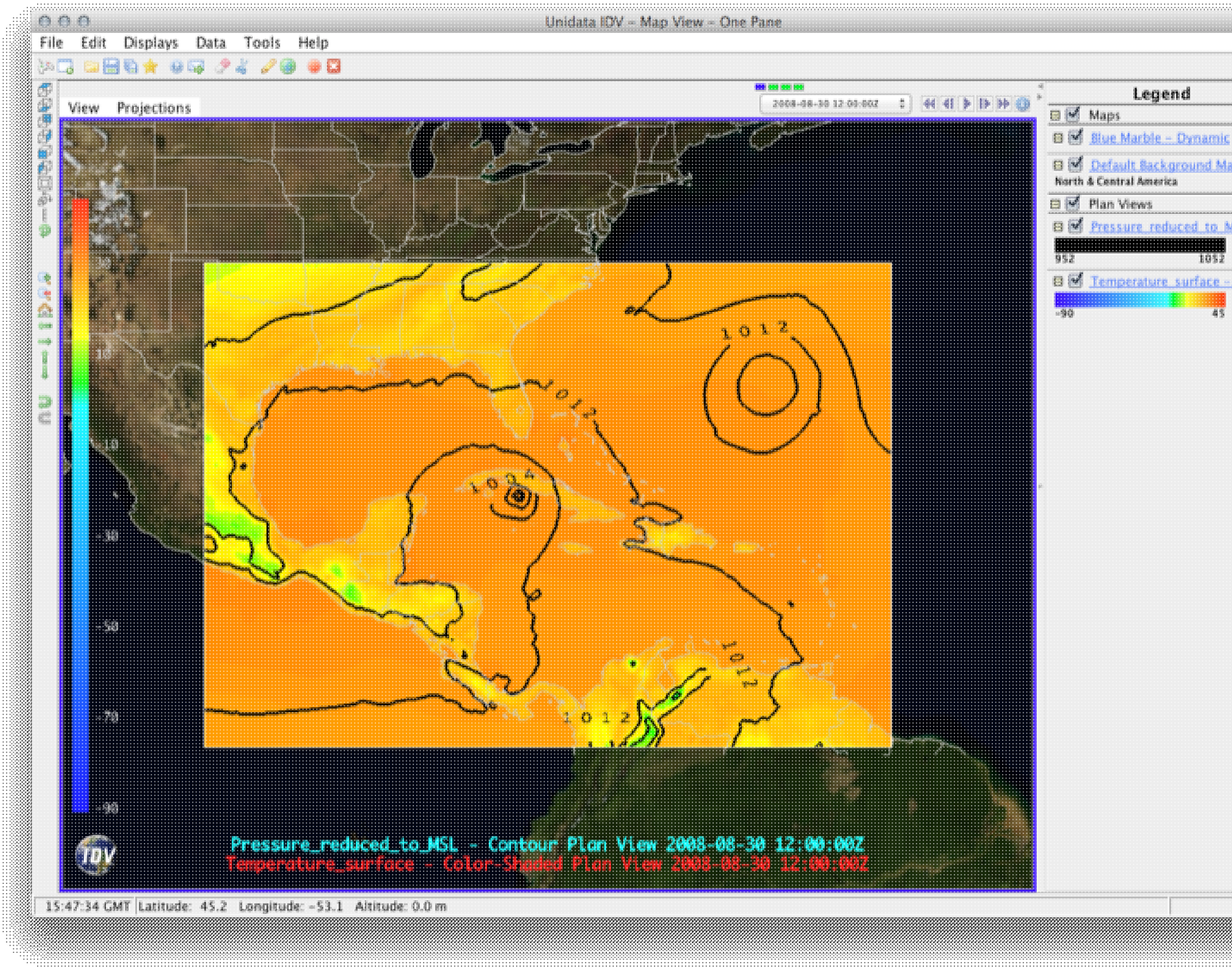
1. From the main menu bar, select **File** **Save As** to bring up the `Save` dialog box.
2. Make sure `Views`, `Displays`, and `Data Sources` are checked.
3. Make sure the `File Format:` is set to `IDV Bundles (*.xidv)`, and name the bundle `basicSetup.xidv`.
4. Save your bundle in the same directory as the WRF output on the memory stick



To test the bundle, click the scissors symbol, found just below the main menu bar, to clear out all of the data and displays. Then:

- From the main menu bar, click **File** **Open**.
- Use the file dialog box to navigate to your bundle, and select your bundle.
- An `Open bundle` dialog box will appear — make sure `Remove and displays & data` and `Try to add displays to current windows` are checked and click `OK`.

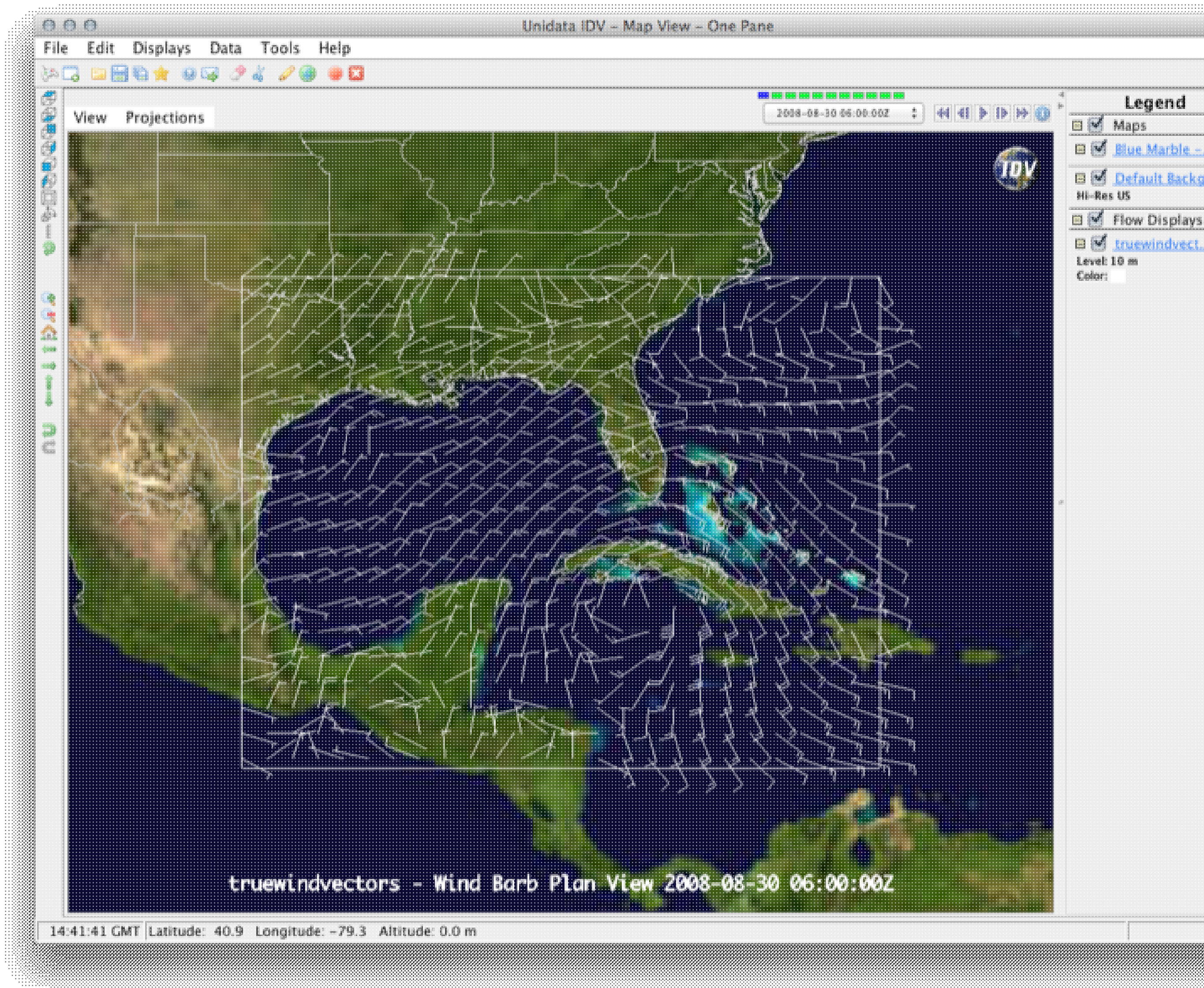
Plan Views



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab. Select `Temperature @ Surface` and the `Color-Shaded Plan View` display, then click the `Create Display` button.
4. In the `Displays` panel, Select `Temperature @ Surface`, and click **Edit Properties**, then select the `Color Scale Tab` and check the box next to `visible`. Other properties of the color scale can be accessed from this tab, such as position and font properties. Click `OK` to apply the changes.
5. Return to the `Field Selector` panel, expand the `2D grid` tab, select `Pressure_reduced_to_MSL @ msl` and the `Contour Plan View` display, then click the `Create Display` button.
6. In the `Displays` panel, select the `Pressure_reduced_to_MSL` field. Click the `Change` box next to Contour to change the contour properties, such as base contour, interval, and label properties.

7. In the `Displays` panel, change the contour color to black by clicking the box next to Color Table (likely labeled `PressureMSL`) and selecting **Solid Black**.
8. Animate your display by clicking the **"play"** button, located just above and to the right of your display on the Main View Window.

Wind Barb and Vector Displays



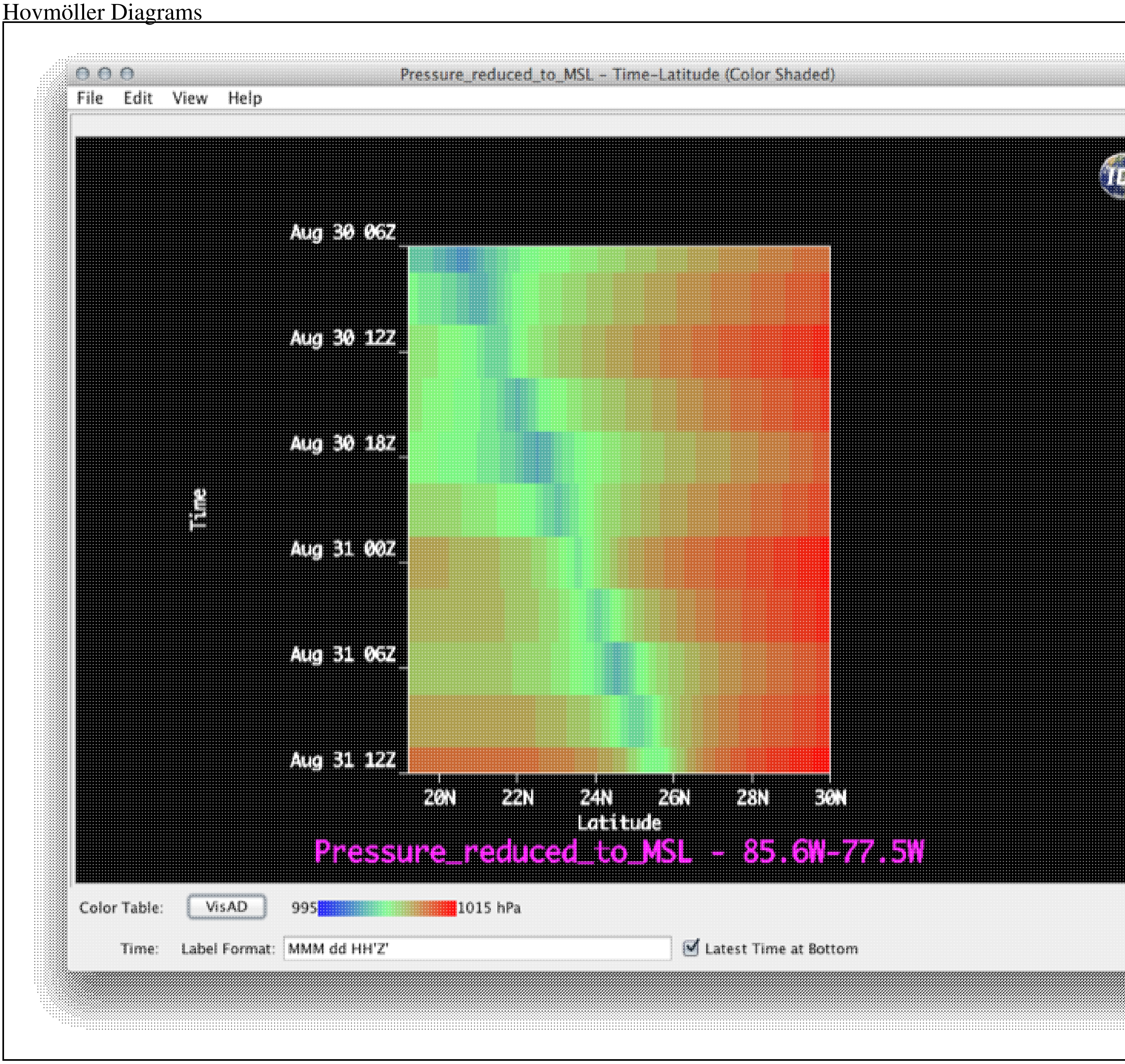
1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab. Scroll down and expand the `Derived` tab. Select the `True Wind Vectors` field and the `Wind Barb Plan View` display, and then click the `Create Display` button.

The initial display of the wind barbs will be quite crowded, so we will need to clean things up a bit.

- In the `Displays` panel, select `truewindvectors` under `View1`, and set the `Skip: XY:` factor to 10.
- Change the color of the barbs to white by selecting "white" from the dropdown box next to Color.

Note that you can also change from a `Wind Barb` view to a `Streamline` view from the display tabs by clicking the button next to `Show: Streamlines`. Also, if you would rather see a vector plot (as opposed to a barb plot), simply select `Vector Plan View`, rather than `Wind Barb Plan View`, from the `Field Selector` when creating the display.

Hovmöller Diagrams



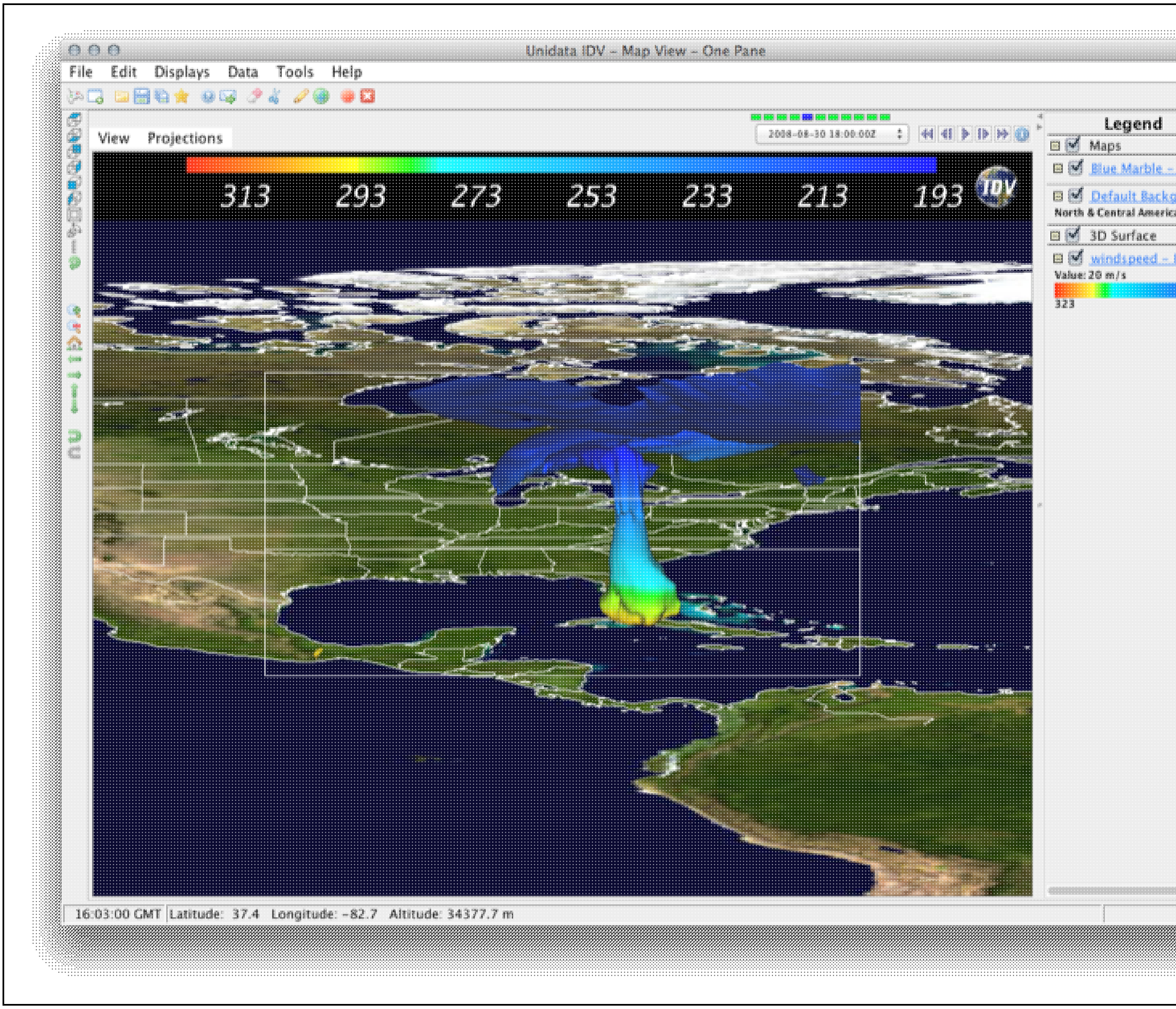
1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab. Select `Pressure Reduced to MSL @ msl`.

In order to clearly see the Hurricane path, we will subset the dataset before we create the Hovmöller diagram.

4. After selecting `Pressure Reduced to MSL @ msl`, click the `Region` tab located in the bottom right panel, uncheck `Use Default` and select the region of interest by using your mouse to click and drag a rectangular selection on the map.
5. Selet the **Hovmoller Time-Latitude (Color Shaded)** display, then click the `Create Display` button.
6. In the `Displays` panel, select the view for the Hovmöller diagram. Then, from inside the `Displays` tab, click **View Undock from Dashboard** to view the diagram in its own window.

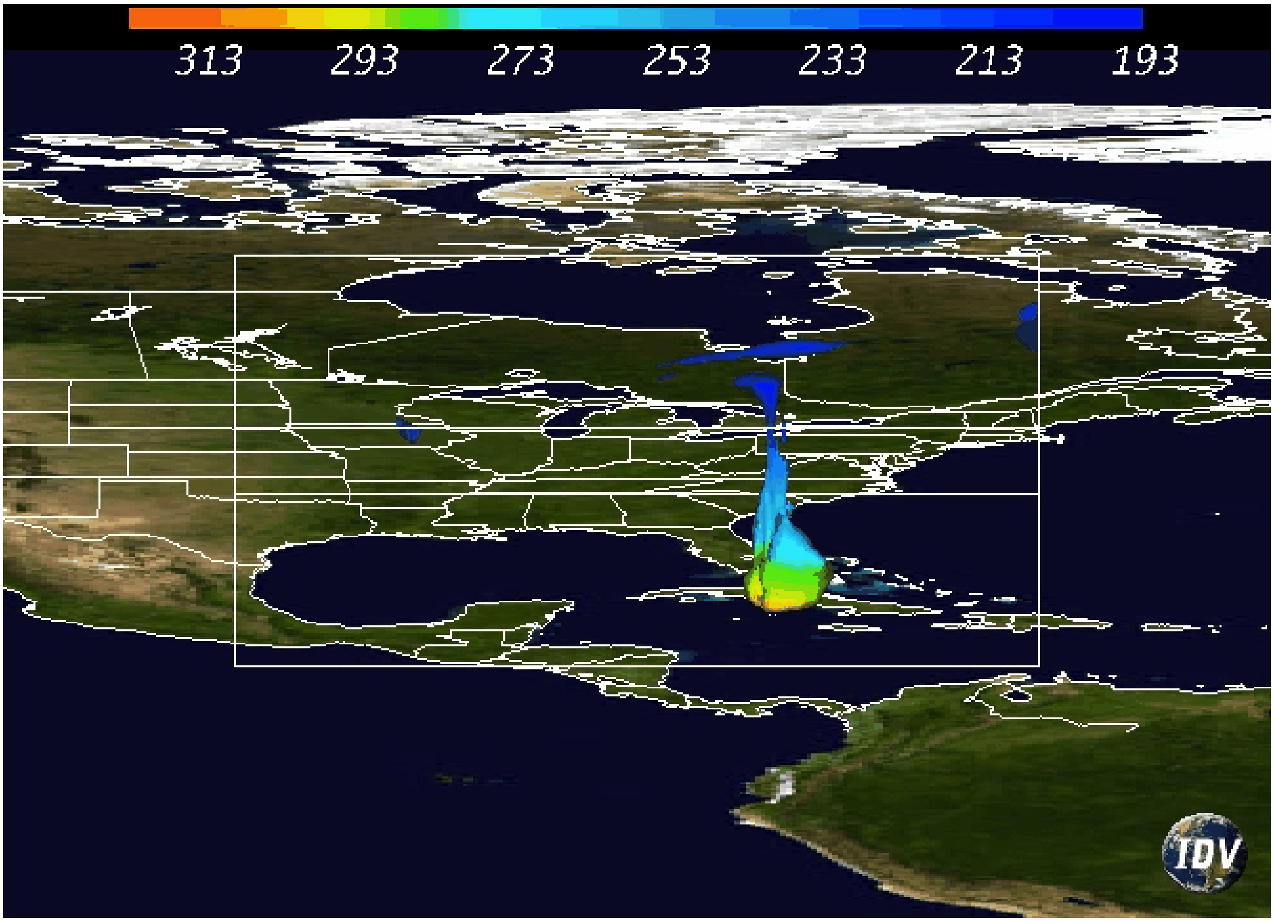
The Time-Latitude Hovmöller display shows Pressure Reduced to Mean Sea-Level as a function of time and latitude. Because we chose `Time-Latitude (Color Shaded)`, mean sea-level pressure will be averaged over longitude for each time-step. To average over latitude instead, recreate the display using `Time-Longitude (Color Shaded)` display.

Isosurfaces

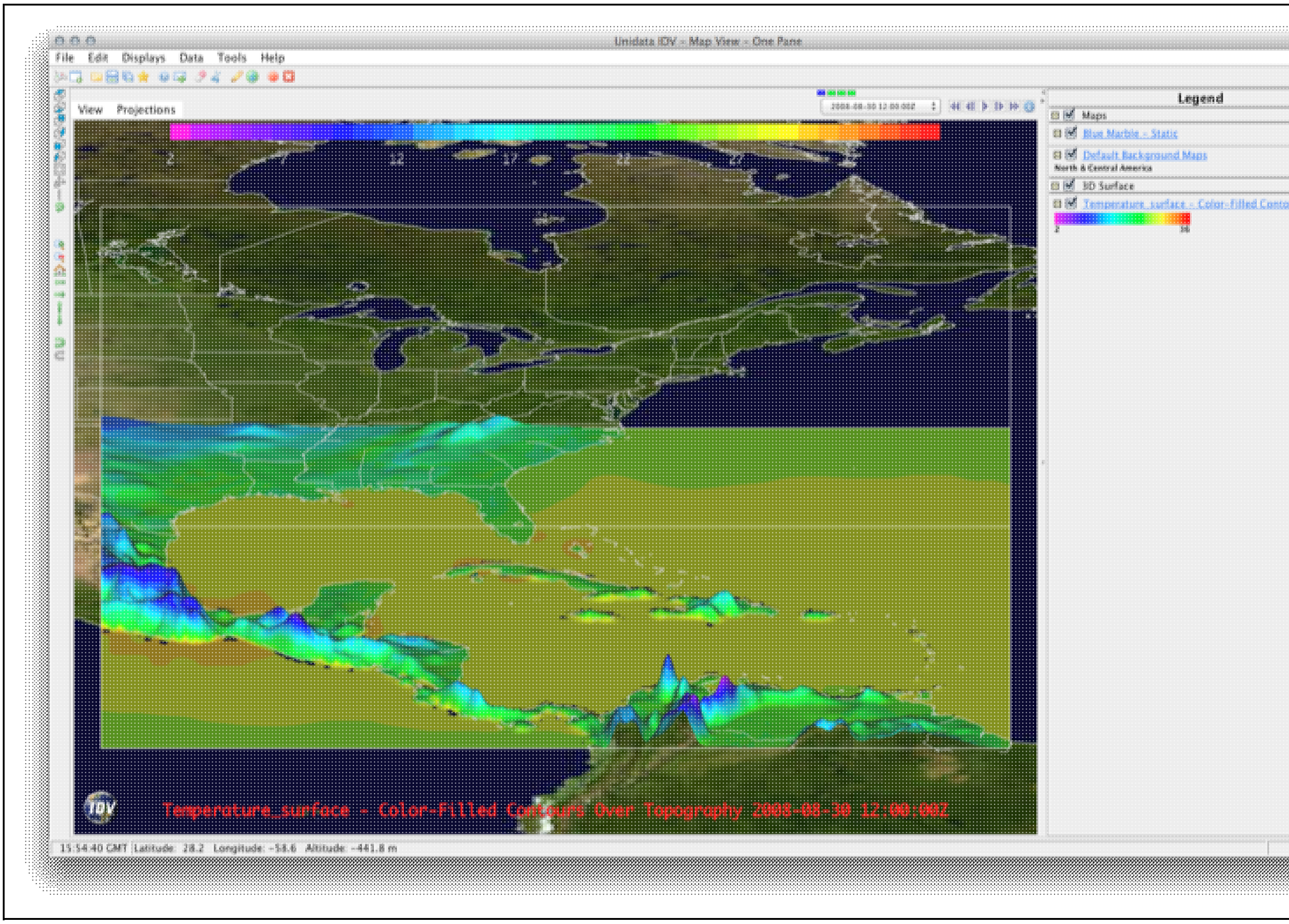


1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is <u>loaded</u> (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `3D grid` tab. Scroll down and expand the `Derived` tab. Select the `Speed (from u_wind & v_wind)` field and the **3D Surface Isosurface Colored by another parameter** display, and then click the `Create Display` button.
4. A pop-up box will prompt you to select which parameter you would like to color the wind speed isosurface. Select **3D grid Temperature @ isobaric**.
5. In the `Displays` panel, set the Isosuface value to 20 m/s.
6. In the Main View window under the Legend List (right edge of the window), right-click the color table and select `Edit Color Table`. This will bring up the color table editor.
7. From the editor window, place your cursor over the color table and right click. Select `Invert Color Table`, and then click `OK` to exit the editor.

8. In the `Displays` panel, Select `windspeed - Isosurface`, click **Edit Properties**, then select the `Color Scale` Tab and check the box next to `visible`.

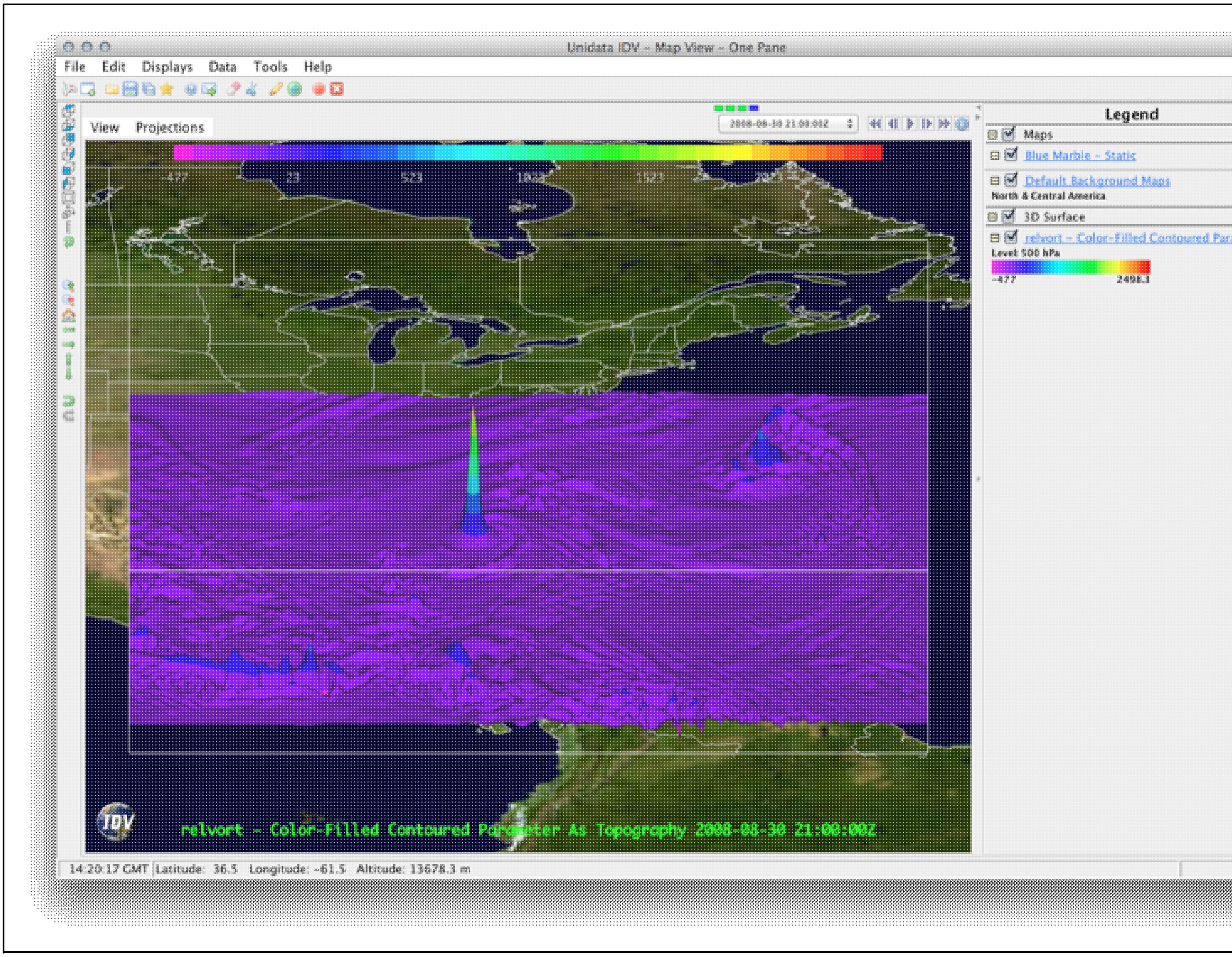


Parameter over Topography

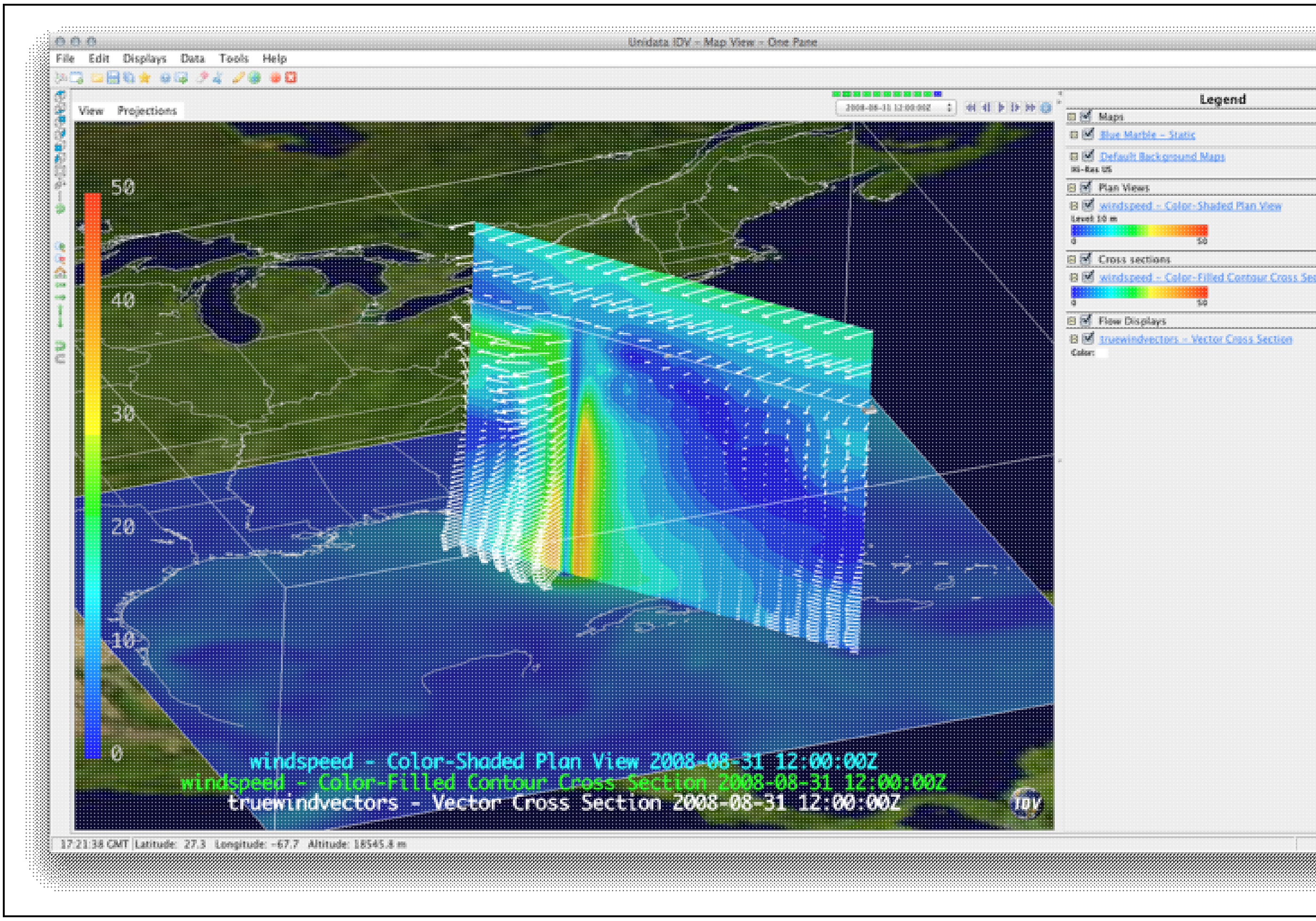


1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab and select the `Temperature @ surface` field and the **3D Surface Color-Filled Contoured Parameter Over Topography** display.
4. A `Field Selector` dialog box will appear and ask you to select the `Topography` field. From your WRF output, select the `2D grid` tab and select the `Geopotential_height @ surface` field and click `OK`.
5. Right click the color scale in the legend (right panel in the Main View Window) and change the color scale by selecting **Basic Bright38**.
6. Again, right click the color scale in the legend (right panel in the Main View Window), but this time select `Change Range...`
7. Click the `Use Predefined` box, select `From All Data`, and then click `OK`.
8. In the `Displays` panel, select `Temperature_surface -...`. Click **Edit Properties**, select the `Color Scale Tab` and check the box next to `visible`. Click `OK` to apply the changes.

Parameter as Topography



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `3D grid` tab. Scroll down and expand the `Derived` tab. Select the `Relative Vorticity (from u_wind & v_wind)` field and the **3D Surface Color-Filled Contoured Parameter as Topography** display.
4. Select the "Level" tab from the subset panel and select 500 hPa, and then click the `Create Display` button.
5. Right click the color scale in the legend (right panel in the Main View Window) and change the color scale by selecting **Basic Bright38**.
6. Again, right click the color scale in the legend (right panel in the Main View Window), but this time select `Change Range...`
7. Click the `Use Predefined` box, select `From All Data`, and then click `OK`.
8. In the `Displays` panel, select `relvort - Color-Filled...` . Click **Edit Properties**, select the `Color Scale Tab` and check the box next to `visible`. Click `OK` to apply the changes.



The image above is created from velocity data. A 2D color shaded plan view of speed at 2 m AGL is displayed along with two cross sections — one of wind speed and the other of horizontal wind vectors. Both cross sections are set to coincide, though this is not the default behavior. To produce the cross sections shown above, do the following:

1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab. Scroll down and expand the `Derived` tab. Select the `Speed (from u_wind & v_wind)` field and the **Cross sections Color-Shaded Plan View** display, and then click the `Create Display` button.
4. Return to the `Field Selector` panel, expand the `3D grid` tab. Scroll down and expand the `Derived` tab. Select the `Speed (from u_wind & v_wind)` field and the **Cross sections Color-Filled Contour Cross Section** display, and then click the `Create Display` button.
5. Return to the `Field Selector` panel, expand the `3D grid` tab. Scroll down and expand the `Derived` tab. Select the `True Wind Vectors (from u_wind & v_wind)` field and the **Flow Displays Vector Cross Section**, and then click the `Create Display` button.

Notice that at this point, each cross section has its own control in the view window. To link the cross sections such that they coincide, do the following:

6. In the `Display` panel, select `windspeed - Color-Filled Contour Cross Section`, then click **Edit Sharing Sharing on**.
7. Repeat the above step for the `truewindvectors - Vector Cross Section` display.
8. Use your mouse to move one of the cross section indicators in the Main View Window to 'snap' the two separate cross sections together.

Now that the two cross sections are linked, let's clean up the vector plot as it's pretty cluttered:

9. In the `Displays` panel, select `truewindvectors` under `View1`, and click the `Settings` tab.
10. Change the color of the barbs to white by selecting "white" from the dropdown box next to Color.
11. Reduce the number of vectors being plotted by clicking **Edit Properties**, select the `Spatial Subset` tab in the Properties box that appears, and set `X Stride:` and `Y Stride:` to `Every twentieth point`.

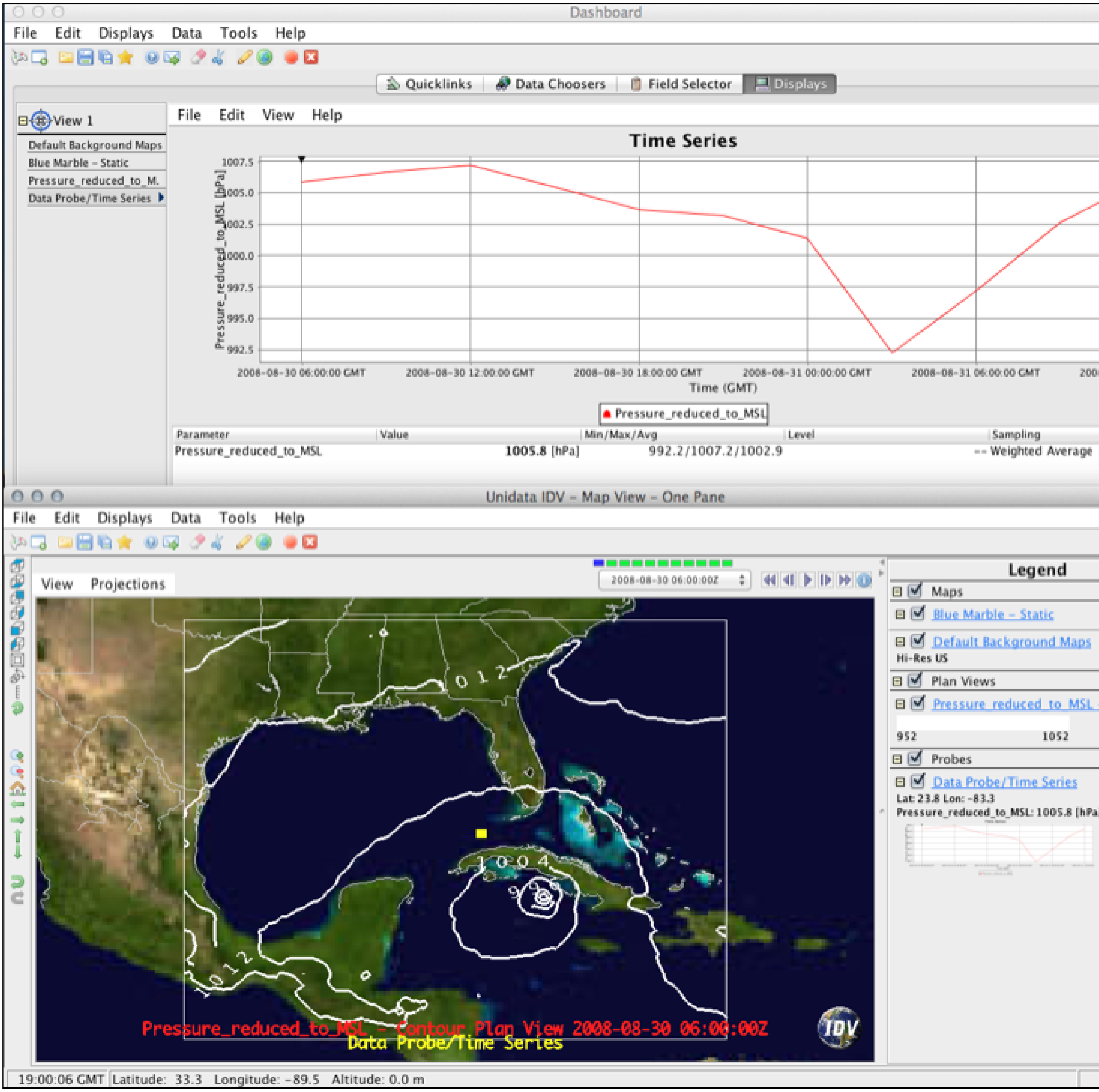
As a final step, make sure the color scale used in both speed displays (plan view and cross section) have the same range, and display one of the scales. For each color scale in the legend (right panel in the Main View Window):

12. Right click the color scale in the legend (right panel in the Main View Window), select `Change Range...`
13. Click the `Use Predefined` box, select `From All Data`, and then click `OK`.

Then, add the color scale to the display:

14. In the `Displays` panel, select either `windspeed -` listed under `View1`, and set the `Settings` tab.
15. Click **Edit Properties**, select the `Color Scale Tab` and check the box next to `visible`. Click `OK` to apply the changes.

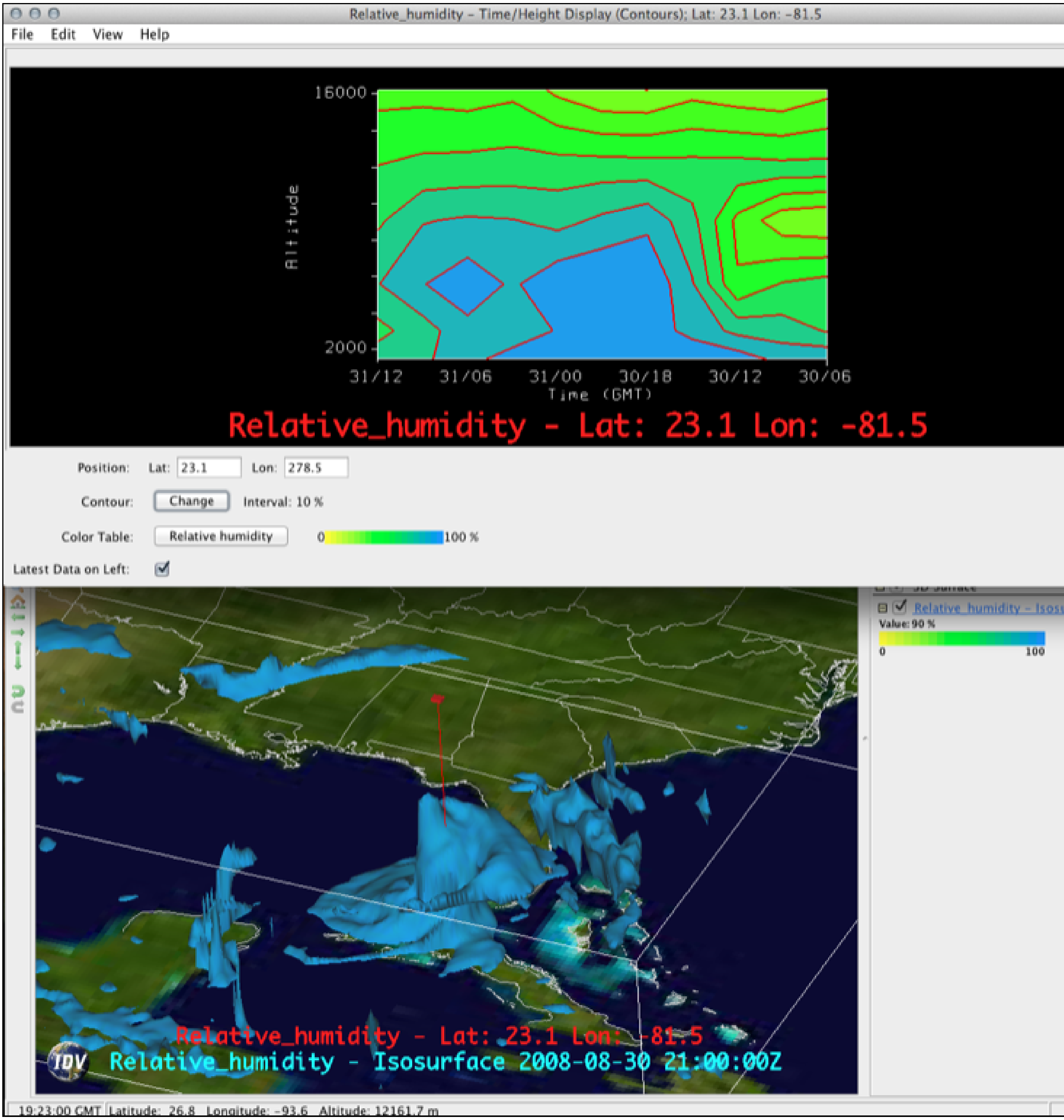
2D and 3D Data Probes



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `2D grid` tab and select the `Pressure_Reduced_to_MSL @ msl` field. Then, hold control (or Command, if on a Mac) and select the **Plan Views Contour Plan View** and `Data Probe/Time Series` displays, and then click the `Create Display` button.

In the Main View Window, you will see contours of `Pressure_Reduced_to_MSL @ msl`, as well as a Probe marker (in the example above, the marker is a yellow rectangle located just NW of Cuba). You can click and drag the probe marker to move it to a location of your choice. The corresponding time series plot will show up in the Dashboard under the `Display` tab (shown above the Main View Window in our example image). Note that the Data Probe will also work with three-dimensional data, and you will be able to move it vertically as well.

Time-Height Display



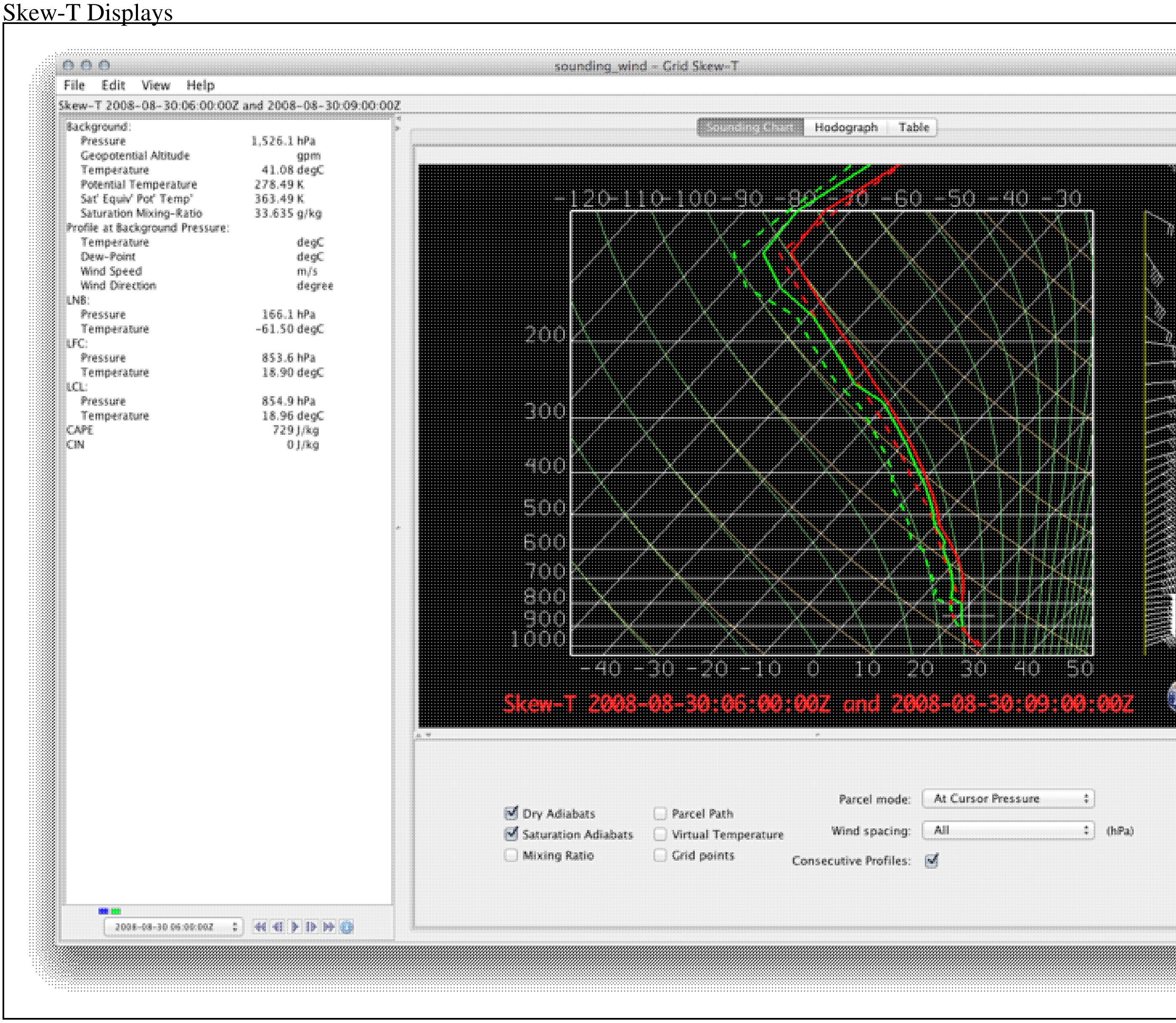
1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `3D grid` tab, and select the `Relative Humidity @ isobaric` field. Then, hold control (or Command, if on a Mac) and select the **3D Surface Isosurface** and **Probes Time/Height Display (Contours)** displays, and then click the

`Create Display` button.

4. In the `Display` panel, select `Relative_humidity - Isosurface`, and set the isosurface value to 90%.

In the Main View Window, you will see the 90% RH isosurface, as well as a time-height marker (in the example above, the marker is red vertical line with a rectangle on top). You can click and drag the probe to move it to a location of your choice.

Skew-T Displays



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded (or utilize the bundle we just created).
3. In the `Field Selector` panel, expand the `3D grid` tab. Scroll down and expand the `Derived` tab. Select the `Sounding Data (with true winds)` field and the `Soundings` -> `Grid`

`Skew-T`.

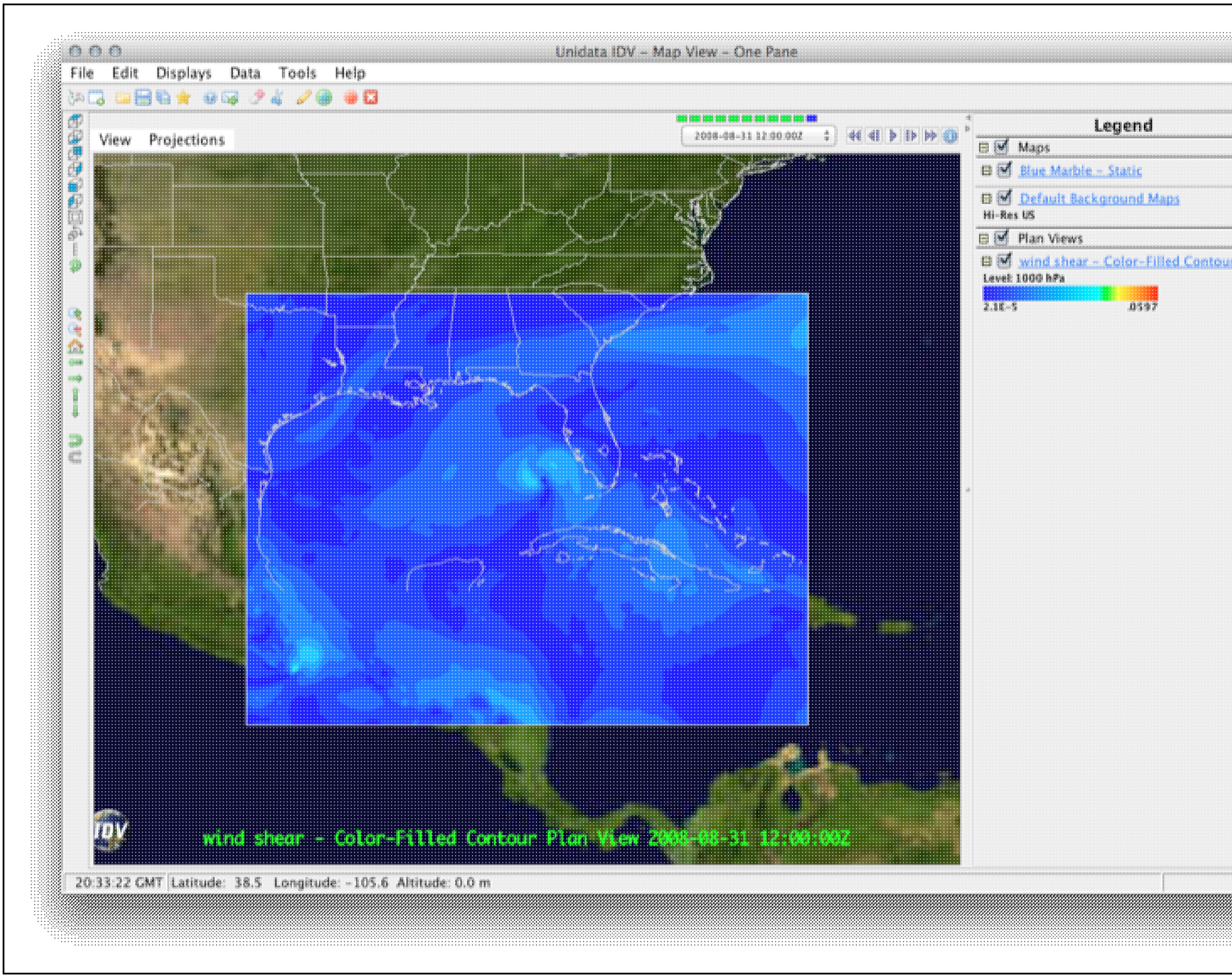
4. Before creating the display, look below the display panel for the subset panel and select the `Times` tab. Select `Use Selected` from the drop-down box and select the first three time steps (hold control or command to select multiple times),then click the `Create Display` button.
5. Next, select the `Region` tab and click-and-drag a rectangular region around Cuba - we're subsetting spatially to focus on the hurricane.
6. Add a Color-Shaded Plan View of the 2D variable `Pressure_reduced_to_MSL @ msl` to more easily navigate the skew-T location marker through the dataset (similar to the time-height marker from the previous example).
7. Place the skew-T location marker in a region of interest and return to the Dashboard `Displays` tab. Undock the display (**View Undock from Dashboard**).

A feature of the skew-T display is that you can animate skew-T's in time. However, sometimes it is nice to see consecutive skew-T's (data from two different times) on the same chart. Underneath the Skew-T diagram are the display controls, which includes the ability to see `Consecutive Profiles` (simply check the box next to `Consecutive Profiles` to enable this feature). To update the sounding parameters, use the middle click on your mouse to set the location of the originating parcel. If you are using a Mac Trackpad, you will need to select a point on the skew-T diagram with **Command+Click**, then use **Option+Click** to update the parameters.

3.13.0.3 WRF Analysis: Formulas

Formulas in the IDV
Built-in Forumlas
The IDV is not only a data visualization package, but it is also a powerful analysis package. The IDV has several built-in functions that range from very simple operations, such as negating a field (that is, changing the sign of all of the values on the grid), to more complex operations, such as computing the horizontal flux divergence field. These built-in operations can be accessed through the `Field Selector` tab by clicking on `Forumlas` in the Data Sources side panel.

As an example, let's use a formula to compute the 500 mb - 850 mb wind speed shear:



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded.

3. In the Field Selector panel, select Formulas from the Data Sources panel and click **Grids Layer Wind Shear**, then select Color Filled Contour Plan View as the display and click Create Display.

The IDV will prompt you for which levels you would like to calculate the shear between (top and bottom), and the fields (u, v, and z) needed to compute the layer shear.

4. For field u, expand the data source for your WRF output and select the **3D grid u_wind @ isobaric** field.
5. For field v, expand the data source for your WRF output and select the **3D grid v_wind @ isobaric** field.
6. For field z, expand the data source for your WRF output and select the **3D grid Geopotential_height @ isobaric** field.
7. Click the Create Display button.

Adding Custom Functions

Formulas in the IDV are defined using Jython, which means extending the analysis capabilities of the IDV can be done by adding new Jython functions (much easier than editing the IDV Java code!) As an example, let's add a Jython formula to compute wind speed (although one already exists, it's still a good exercise to go through):

1. Remove all displays.
2. From the main menu, click on **Edit Formulas Create Formula**. The Formula Editor window should appear.
3. Enter the Name of your forumla: windspeed (one word - no space).
4. Enter the mathematical formula definition in the Formula entry field using the Jython syntax. The formula is
```
sqrt(u**2 + v**2)
```
5. Open the Advanced panel in the Formula Editor.
6. Enter a Description for the forumula: wind speed from u and v.
7. For Group enter Workshop.
8. Now we will tell the IDV which display types our formula can use:
    - In the Displays section, click the Use selected radio button, then click the All off button.
    - Expand the Plan Views category and check the Contour Plan View option. Also, expand the 3D Surface category and check the Isosurface option.
9. Click on Add Formula.

A new item wind speed from u and v should appear in the Field Selector window's Fields panel, under Workshop. We've only touched on the basics of the use of formulas in the IDV - please see the IDV Formulas for more details.

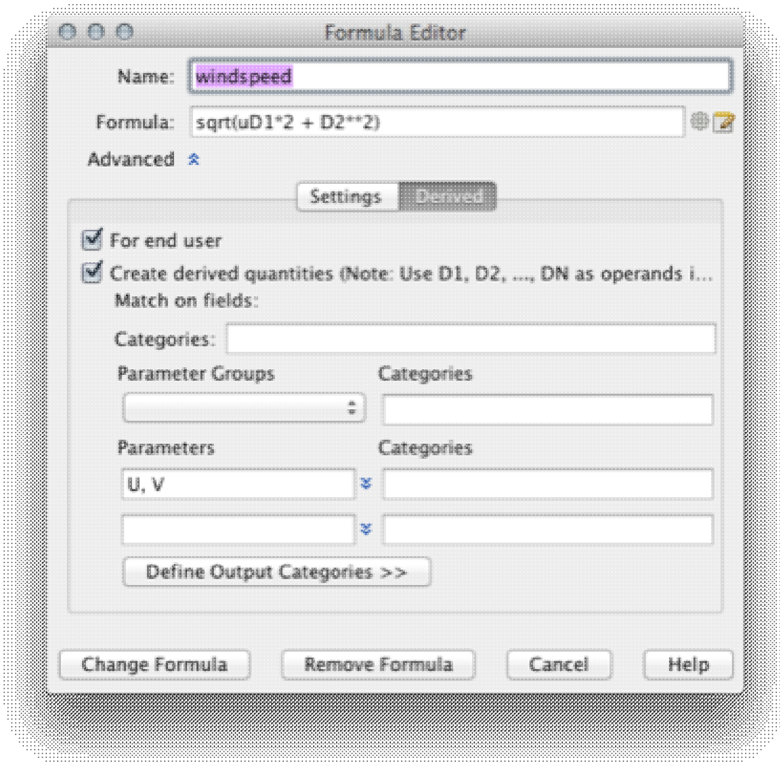
3.13.0.4 WRF Analysis: Derived Products and Advanced Features

Derived data products and advanced features.
Derived Data
The IDV has the ability to make quantities computed through the use of formulas available under the loaded data source rather than going through the steps to use a formula, as outlined in the previous section. For example, you could compute wind speed using the formula directly — simply load a Data Source, then from the Field Selector click **Formulas Workshop Windspeed from u and v data**, then click `Create Display`, then choose the fields for `u` and `v` and THEN you're done. Wouldn't it be nice if the IDV could recognize the `u` and `v` fields in your dataset and automatically present you with the choice to display `Flow Vectors` from `u` and `v` next to the available fields listed in your dataset? Well, this is exactly what Derived Quantities in the IDV accomplish.

Derived quantities can be created by doing the following:

1. Defining an IDV formula by name, description, and mathematical formula.
2. Using special operators (D1, D2, ... DN) for the variable names.
3. Specifying the binding between the special operators and the parameters names or system aliases.
4. Saving the formula and loading or reloading a data source.

In the last section we discussed how to create a new Jython formula for wind speed, so the first step is complete. Now, let's examine how to turn our wind speed formula into a derived quantity. Let's open our wind speed function in the Forumla Editor by selecting the **Edit Formulas Edit Formulas Workshop wind speed from u and v** menu item. This brings up the `Formula Editor` dialog box we previously used to edit a formula. In this box, click the `Derived` tab to begin the process of enhancing the wind speed formula to produce derived quantities:



1. Click the `Derived` tab.
2. In the `Formula`, change u and v to D1 and D2, respectively. Note that the order of the parameters listed in the Parameters box is the same as the order used in the formula (i.e. D1 is U, D2 is V).
3. In the Parameters box, select the drop down list on right side and choose **Aliases Group #1 U component of wind (U)** and **Aliases Group #2 V component of wind (V)**. Note that Group #1 and Group #2 do not correspond to the 1 and 2 in `D1` and `D2`. The "1" and "2" in the `D#` variables are associated with the order of the parameters listed in the Parameter box; using groups in the alias list helps keep the list from becoming too long to display in the GUI.
4. click `Change Formula`

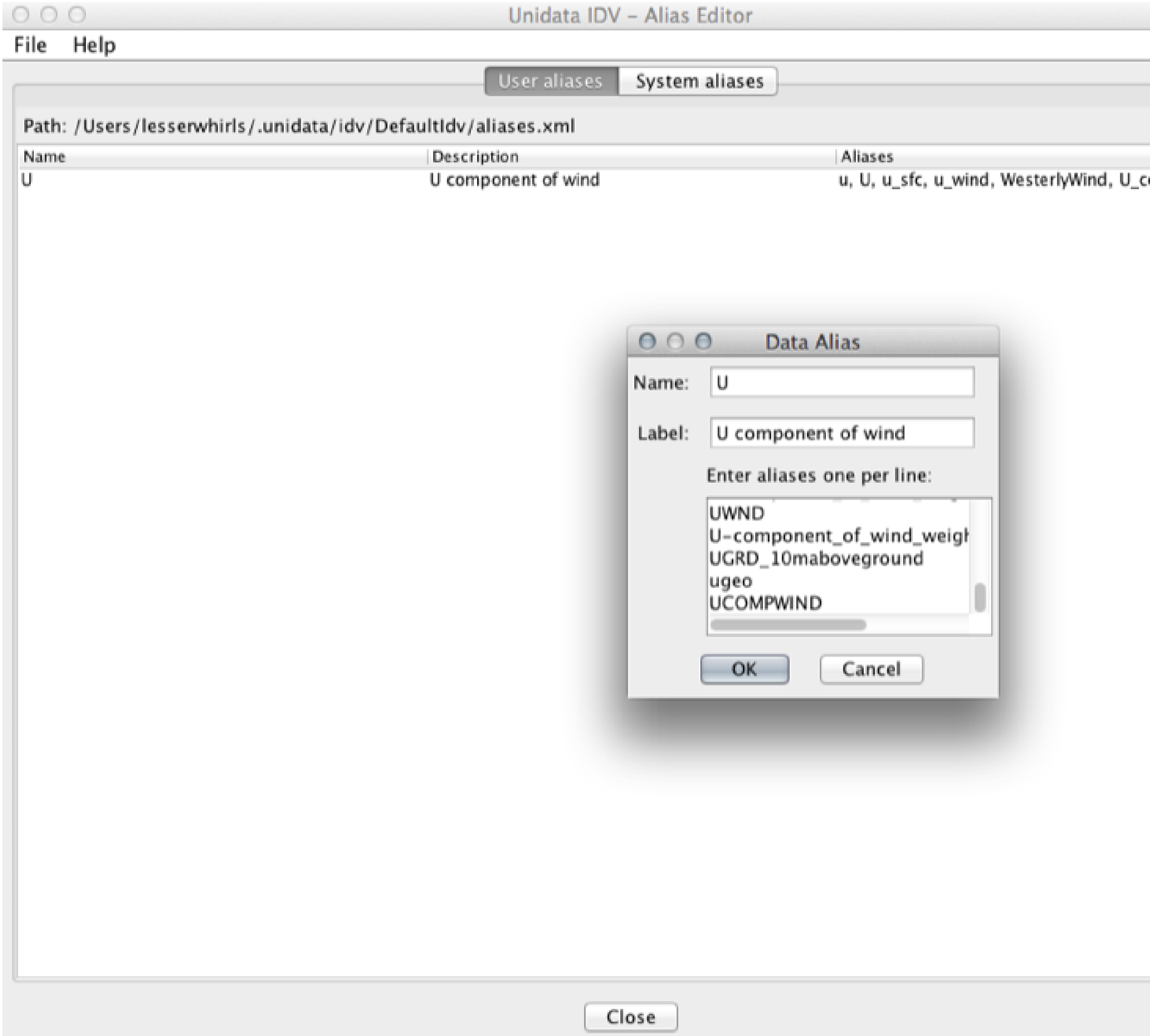
If we reload our data sources (right click on the data source in the Filed Selector and click reload), we will see a new category of data listed in the fields panel called `Workshop`, which when expanded contains our wind speed function.

There are other properties you can assign to your derived quantity, but this is the minimum effort needed to get things working (for more information, see Derived Products in the Users Guide).

Aliases

Sometimes the IDV will not recognize the variables in your file, and thus will not be able to associate D1 with U because it does not know the name of the u component of velocity in your data set. In order to get the IDV to recognize the variables in your dataset, you may need to edit the `Parameter Aliases`. Click **Tools Parameter Alias Editor**, then select `System Aliases`. Here you will find a list of Names (which are those used in the `Parameters` box in the Formula Editor), descriptions, and variable names associated with each Name.

For example, let's say you have a dataset in which the u component of velocity is called 'UCOMPWIND'. If we double click on the "U" name, the Alias Editor will automatically add a copy of the "U" alias to our `User Aliases` list and prompt us to add a new name to the alias list.

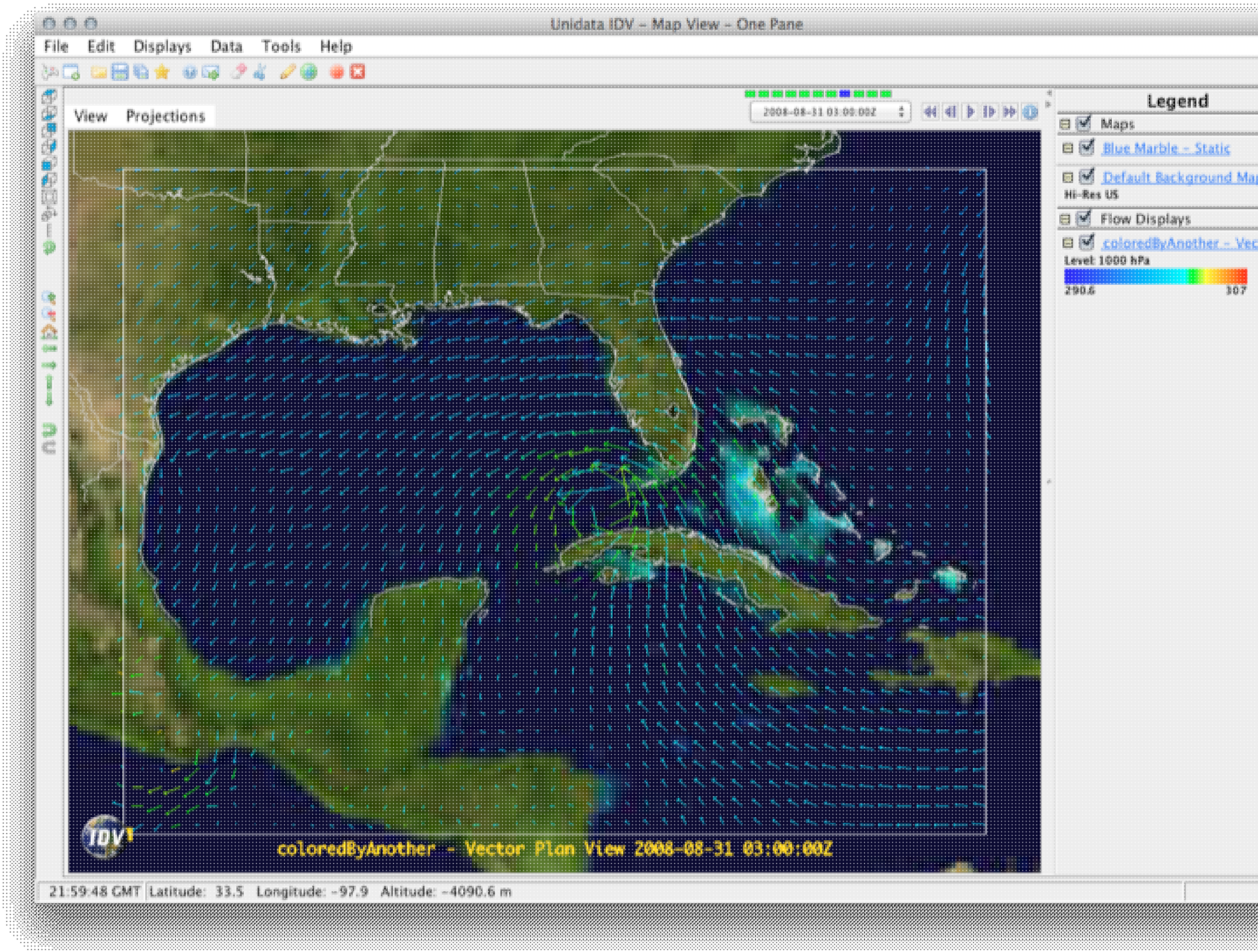


Once UCOMPWIND is added to the alias list, the IDV will know to associate UCOMPWIND with the U variable used in derived quantities. Changes to the alias list also apply to the system derived quantities, such as:

- Dewpoint
- Mixing ratio
- Relative Humidity
- Dewpoint Depression

- Equivilent Potential Temperature
- Potential Temperature
- Horizontal Divergence
- Relative Vorticity
- Height from Geopotential
- Isentropic Potential Vorticity
- Geostrophic Wind Vectors
- Ageostrophic Wind Vectors
- Wind Speed
- 3D Flow Vectors
- True Wind Vectors
- Ocean Flow Vectors
- Sounding Data (with and without winds)

Parameter colored by another
The IDV allows for one parameter to be colored by another parameter's value through the use of a formula.
For example, let's view wind vectors colored by temperature:



1. Clear any existing displays by clicking the red and white eraser symbol found just below the Main Menu bar on either the Dashboard to Main View window.
2. Make sure your WRF ouput is loaded.
3. In the `Field Selector` panel, select `Formulas` as the Data source and expand the `Grids` tab. Then, select the `One value colored by another` formula, choose the **Flow Displays Vector Plan View** display, and click `Create Display`
4. For the `data field` choice, click on your WRF output and select **3D Derived True Wind Vectors (from u_wind & v_wind)**
5. For the `Color field`, select **3D Temperature @ isobaric** and click `OK`
6. From the `Displays` tab, select `coloredByAnother - Vector Plan View` and set the `Skip: XY:` factor to 5
7. Click the button next to `Color Table` and select `Change Range`. Click the `Use Predefined` button and select `From Displayed Data` and click `OK`.

You can change the level at which the colored vectors are displayed directly from the `Displays` tab using the `Levels` drop-down box.

2D and 3D Trajectories (and more about bundles)
The IDV can generate 2D and 3D trajectories. To access the trajectory features:

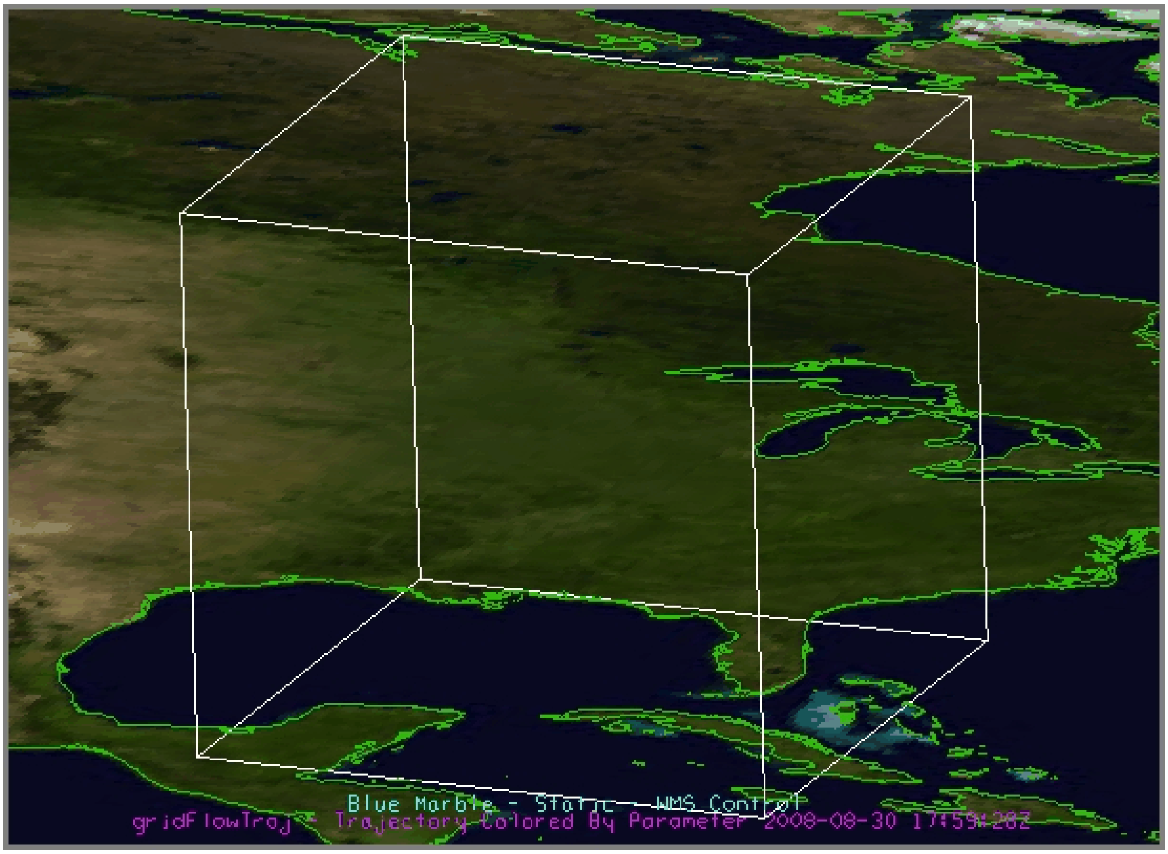
1. Load your WRF data
2. In the `Field Selector`, select the `Formulas` data source and click **Grids Create 3D Flow Trajectory**
3. Next, you will be prompted to select grids for u, v, w, s, and s0)

The fields of `u`, `v`, and `w` are the individual components of the wind. The field `s` is the field you would like to use to color the trajectory. The field `s0` is special — the field itself isn't actually used, but the spatial subset defined in its Region tab **is** used. As the resources used to hold the data for trajectory computation are quite large, it's useful to spatially subset where you would like the trajectories to initiate. A good way to go about this is to use the same field for S and S0, but use the Region tab to subset S0 over the area of interest. Note that this is a new feature in the IDV and the user interface is likely to change!

That said, **loading the data and computing the trajectories can be quite time consuming**, so we've created an IDV bundle that will allow you to explore the trajectory capabilities of the IDV. The bundle can be found on the memory stick given to you as part of this course. However, it can also be found on our RAMADDA server if you are working through this tutorial at home.

Use the IDV to load the GustavTrajectory.zidv bundle (located under the `bundles` folder on your memory stick) and explore the trajectory facility of the IDV. Note that it takes about 3 minutes on a high-end machine to create the display, so please be patient!

- From the main menu bar, click **File Open**.
- Use the file dialog box to navigate to your bundle, and select your bundle.
- An `Open bundle` dialog box will appear — make sure `Remove and displays & data` and `Try to add displays to current windows` are checked and click `OK`.



Ensemble data

The IDV contains many functions designed for analysis of ensemble model data. These functions can be utilized if the data you are using contains an ensemble dimension (alongside temporal and spatial dimensions). The ensemble functions can be accessed in the `Field Selector` panel by choosing "Formulas" as the Data source and expanding the `Grids` tab; The ensemble functions will be listed under the Ensemble tab.

Current ensemble functions include:

- Average over members
- Maximum over members
- Minimum over members
- Mode of the members
- Standard deviation of members
- Percentile calculation
- Range of values
- Univariate probability

The subsetting feature of the IDV can also be used to include/exclude specific members from these calculations.

3.13.1 Jython

3.13.1.0 The Basics

Creating basic WRF displays with Jython and the IDV
The IDV and associated Jython scripting environment is capable of reading **post-processed** WRF output. Specifically, the output need to be on a non-staggered for optimal performance. Also, please understand that the Jython API for the IDV is under active development and calls using Java code must be used to temporarily fill in the gaps. Also, a fair amount of the documentation, such as lists of valid input for functions, can be found in either the IDV java source code or .xml documents; links to these files (hosted on the Unidata github site are provided throughout this part of the tutorial. That said, several analysis functions and basic I/O routines are exposed via Jython and are documented.

If you have participated in the WRF tutorial, then you are likely familiar with the output from the Hurricane Gustav simulation, which is what we will use in our example (although any WRF post-processed output should do fine).

The first step to working with the IDV and WRF output is to load the data. Before we load the data, let's take a moment to create a few helper variables to simplify changing datasets in the future:

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
```

Variable	Usage
`filename`	name of data file (use full path to file or the path relative to the location where the script will run)
`file_opener`	Type of data file

The lines in the above code snippet hold the basic information needed to open a file. The variable `filename` is the name of the WRF file to be opened; `file_opener` must point to the `id` of one of the datasource types the IDV knows — a list of these can be found in datasource.xml (note that the "label" properties in datasource.xml may look familiar, as these names are used in the IDV data chooser). In our example, a GRIB file is what we wish to open, which coresponds to the `id` "file.grid" (the `label` is "Grid files (netCDF/GRIB/OPeNDAP/GEMPAK)"). Now that the basic information about the file has been defined, we can create a dataSource using the Jython method `makeDataSource`.

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
```

Note that `makeDataSource` returns a `DataSource` object, which we've named `ds` in our example.

Now we need to make some decisions regarding which data and display types we would like to use. As in our first step, let's create a few helper variables to hold this information:

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
```

Four new variables were added to the example:

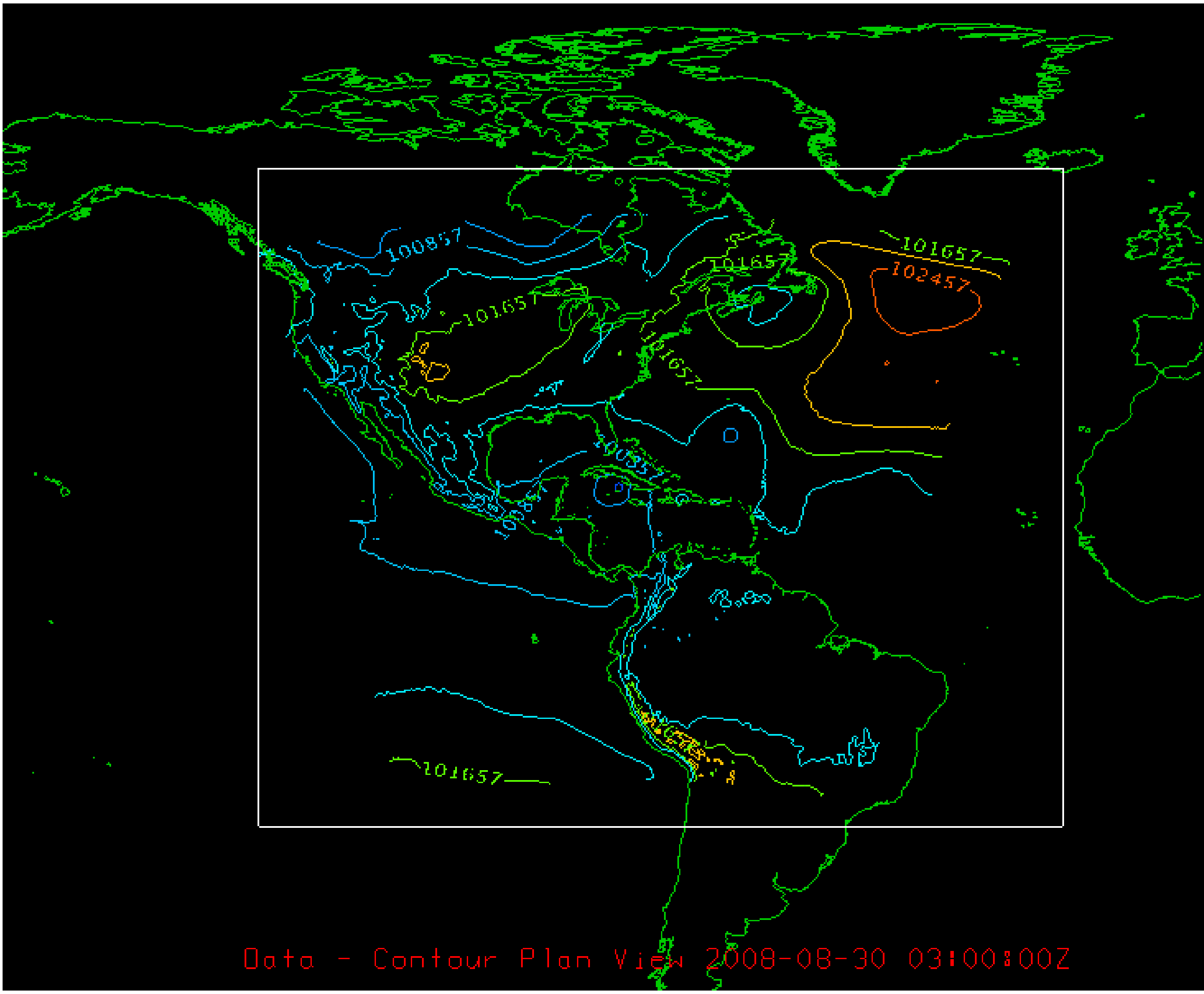
Variable	Usage
`variable`	Data variable to be displayed
`display_type`	Type of display to produce
`image_dimensions`	Dimensions of the output image
`output_name`	Name of the image file to produce*
*note: the format of the image is controlled by the extension used, e.g. .png)	

`display_type` must be one of the types that the IDV knows — these can be found in controls.xml. controls.xml contains a listing of the display controls available in the IDV. Each control has various properties, some of which you may be familiar with if you've used the IDV (for example, the `label` property); however, for the purposes of Jython scripting, we need to use the `id` property. In our example, we are going to create a "Contour Plan View" display (the `label` property) which has the `id` 'planviewcontour'.

We are now at the point where we can generate an image with a few additional lines of Jython:

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
createDisplay(display_type, pressure)
pause()
image = getImage()
writeImage(output_name)
```

The resulting image will look something like this:



It's not the prettiest, but it does show the data. However...

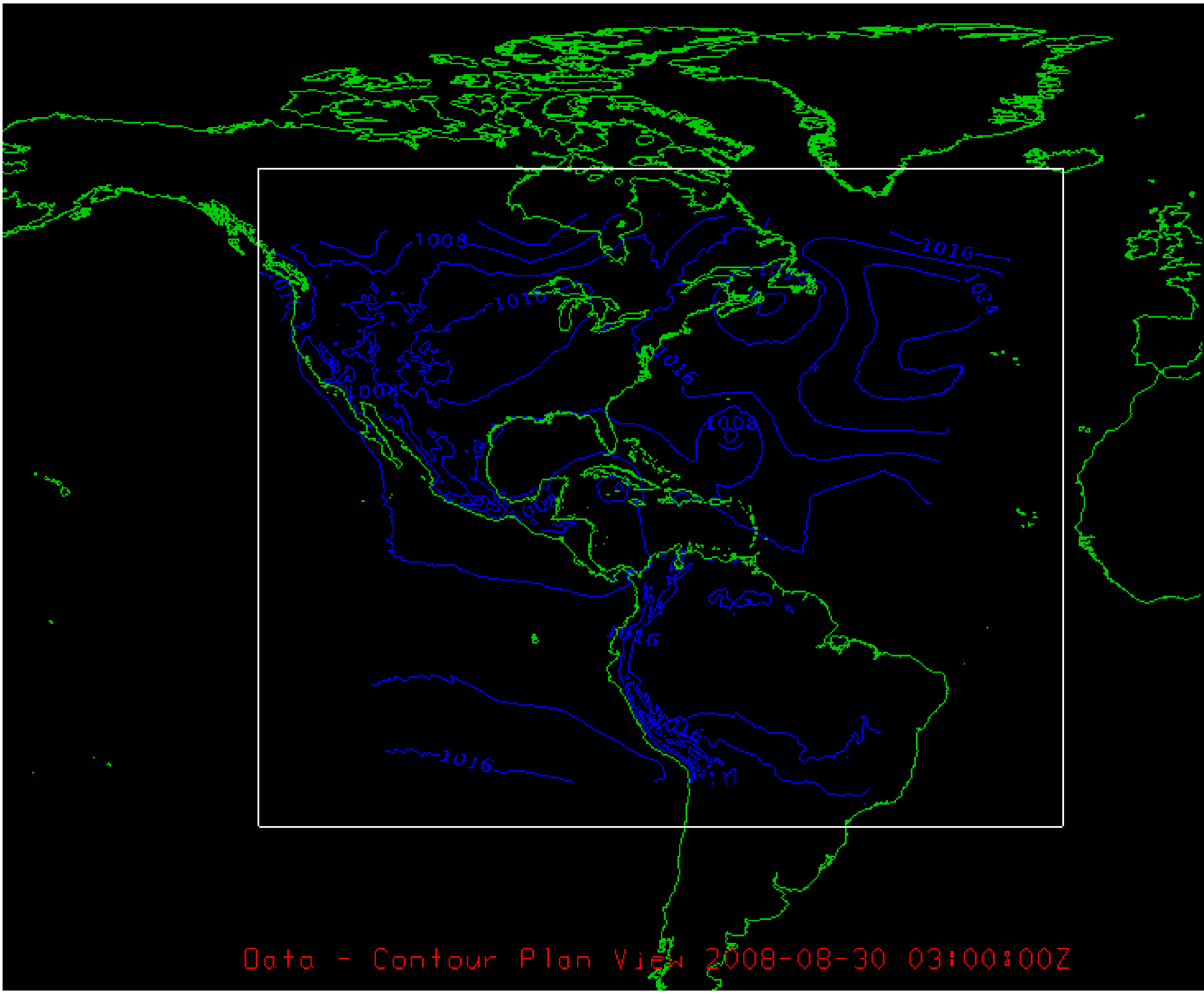
1. It appears the units are in Pa - what if I want them in mb?
2. I would like to change the color and interval of the contours.
3. What is up with the legend (Data?)? Can I add a color scale?

Let's take a moment to address these issues one at a time.

3.13.1.1 Units

How to change the units shown in a display.
Changing units on a display can be done by *modifying* one line of code and adding one new line:

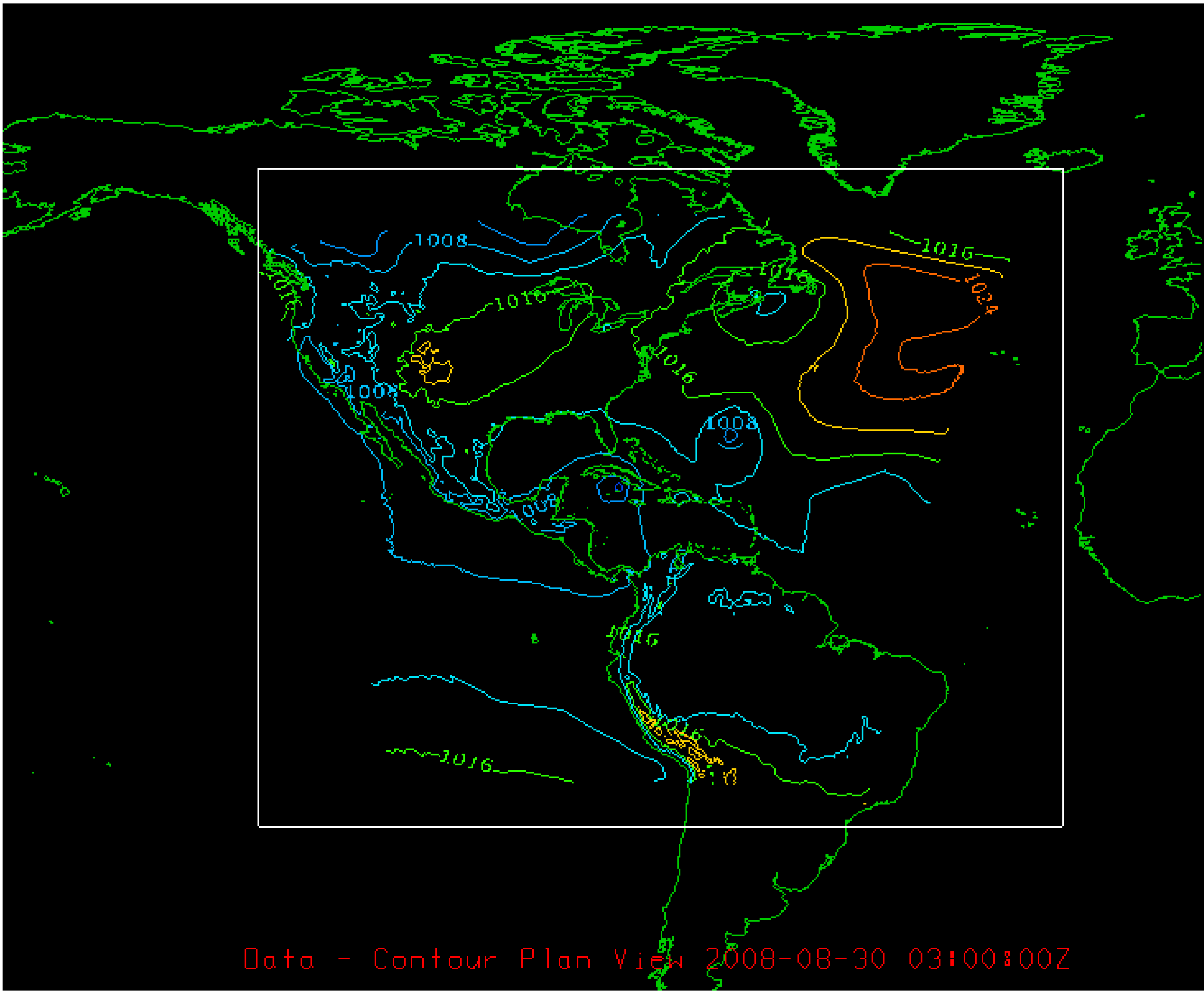
```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
dc.setDisplayUnitName('mb')
pause()
image = getImage()
writeImage(output_name)
```



Note that we've created a `DisplayControlBase` object `dc`(specifically a `ContourPlanViewControl` object, a subclass of `DisplayControlBase`), and changed the unit using the method `setDisplayUnitName`. While the goal of changing units has been accomplished, notice that it seems as though we have lost the characteristic of the contours to be colored based on value. However, this is not the case - we simply need to update the color range, as it still reflects the range of values in units of Pa:

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
```

```
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
old_unit = dc.getDisplayUnit()
dc.setDisplayUnitName('mb')
new_range = dc.convertColorRange(dc.getRange(),old_unit)
dc.setRange(new_range)
pause()
image = getImage()
writeImage(output_name)
```

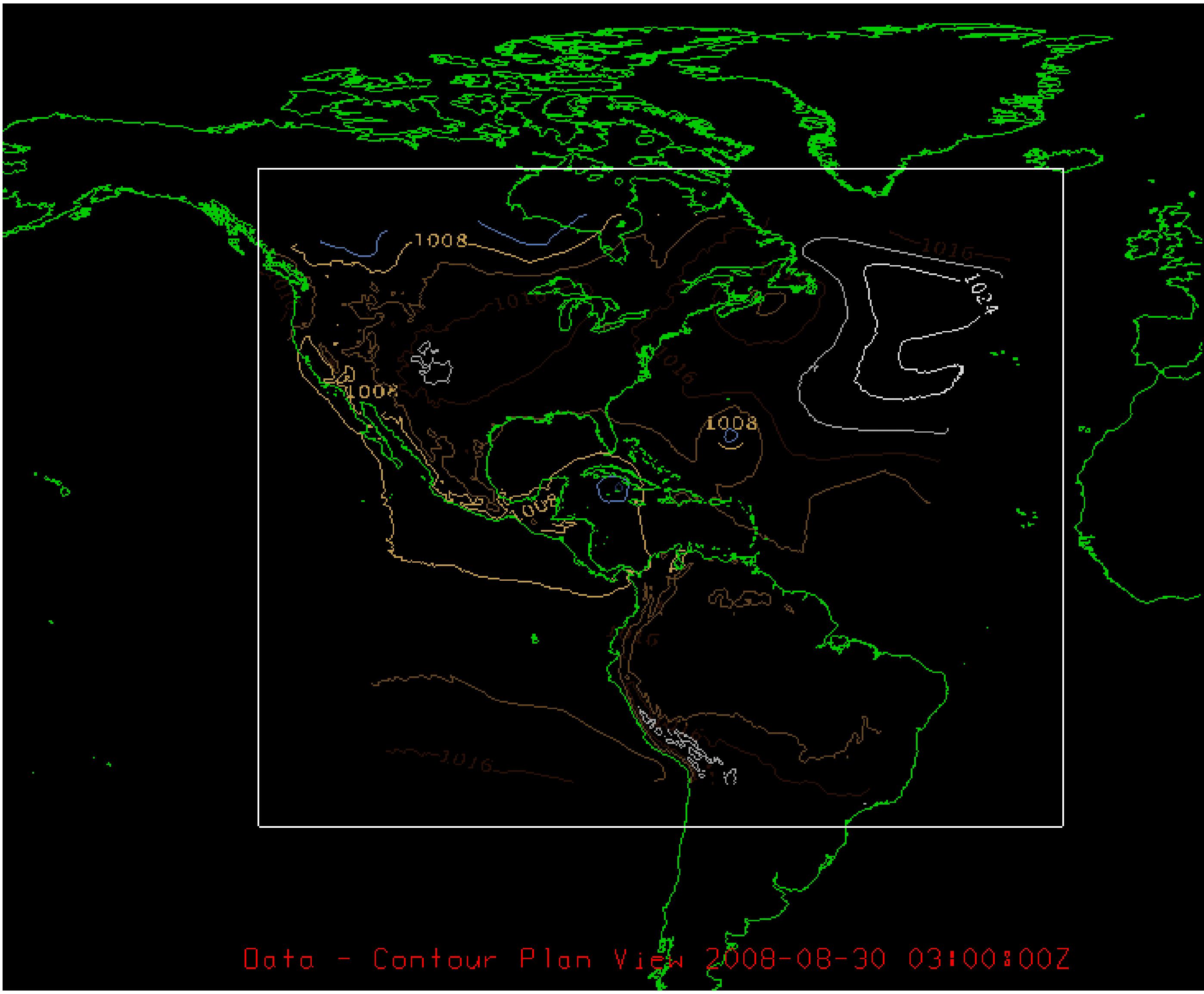


The first thing needed is to save the `old_unit` (Pa), which has to be used in the function to get the `new_range` (the range after converting the display unit to mb). Finally, we set the color range to `new_range` using the `setRange` method, and voil&agrave!

3.13.1.2 Contours

How to extend the basic display of WRF data to change the default contour characteristics.
Let's start off by examining how to change the color of the contours on our image. There are two basic coloring schemes that we can use to do this — use a range of colors from a colormap in which the value of the isoline is reflected by color, or use one solid color, although both will be done in the same way. The first thing we must do is ask the IDV to give us access to the available color table resources. To do this, we must get the current ColorTableManager from the main idv instance `idv` (see `DefaultIDV` for more information). Once we have the ColorTableManager, we can set a new color table for our display. Note that for the built-in colortables, the name comes from the `name` property of the ColorTable object, which can be found in two locations: colortables.xml and extra.xml. These can also been seen from within the IDV using the Color Table Editor. If you've installed other color tables via plug-ins (or created your own), then going through the Color Table Editor is the best way to see a list of the available names. For our example plotting MSL Pressure, the IDV default color table is "PressureMSL". Let's change that to the "TOPO/Sat Composite" color table, just for fun.

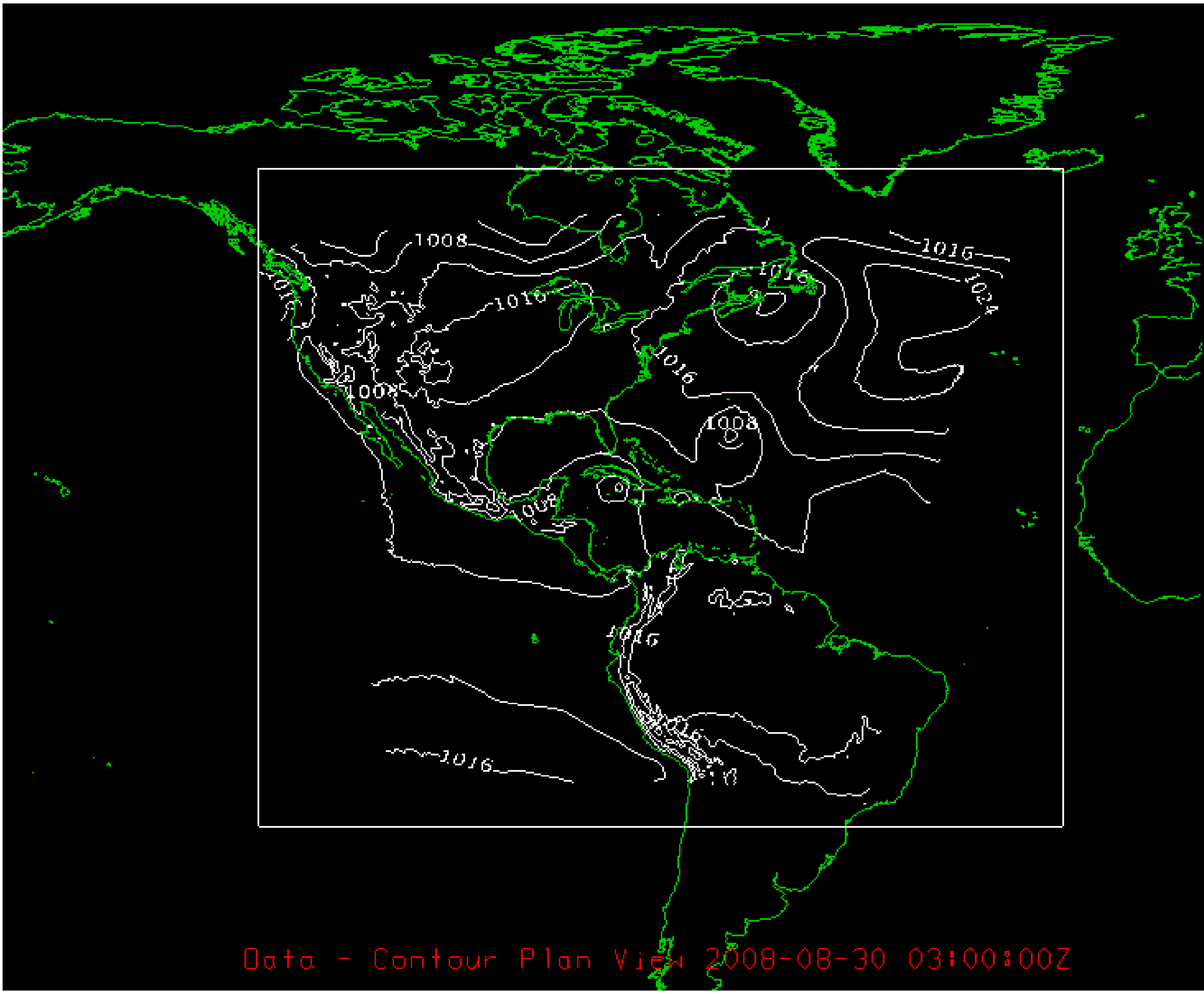
```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
ctm = idv.getColorTableManager()
dc.setColorTable(ctm.getColorTable('TOPO/Sat Composite'))
old_unit = dc.getDisplayUnit()
dc.setDisplayUnitName('mb')
new_range = dc.convertColorRange(dc.getRange(),old_unit)
dc.setRange(new_range)
pause()
image = getImage()
writeImage(output_name)
```



If we would rather have a solid color, say white, then use the Color Table name "white":

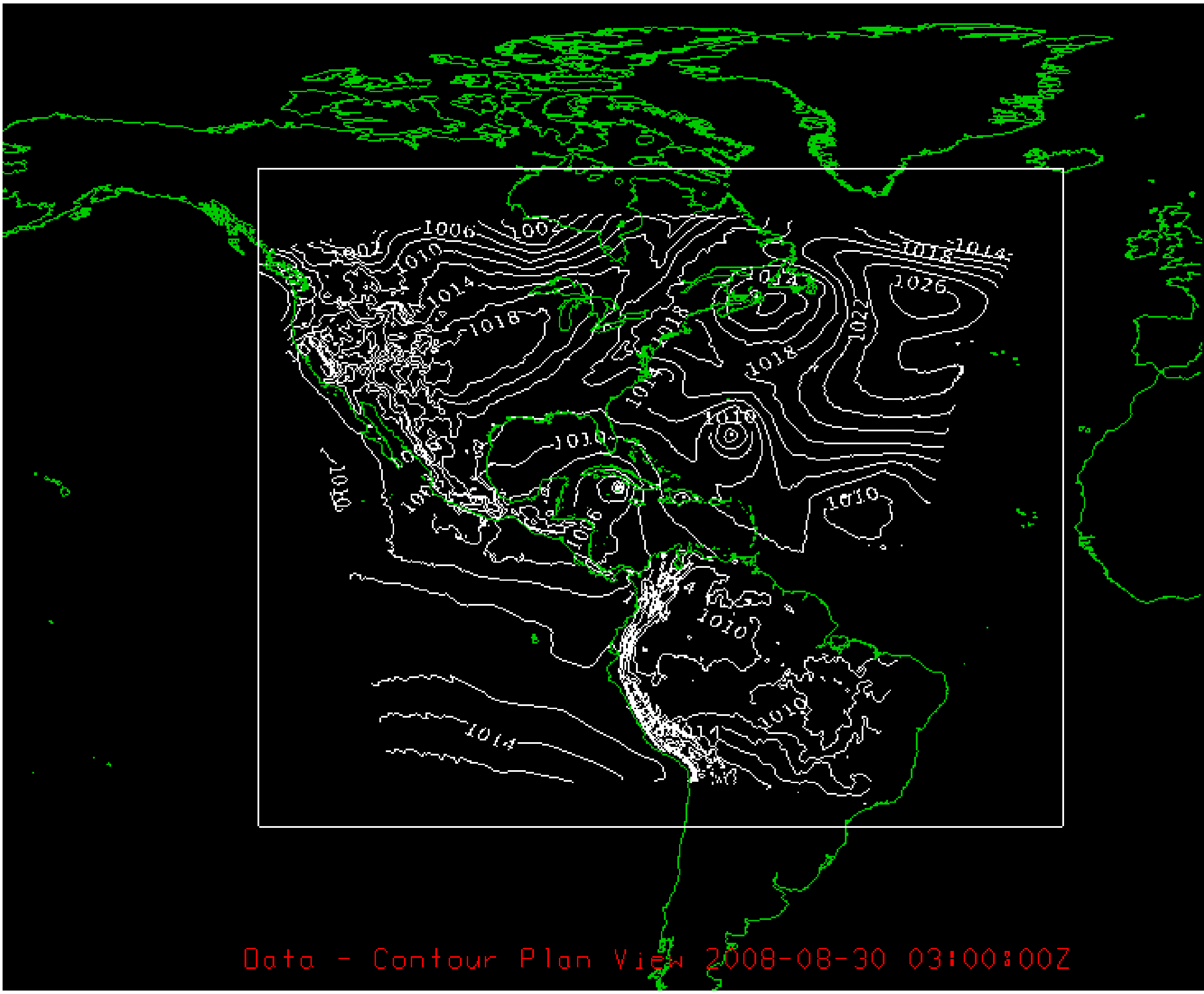
```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
ctm = idv.getColorTableManager()
```

```
dc.setColorTable(ctm.getColorTable('White'))
old_unit = dc.getDisplayUnit()
dc.setDisplayUnitName('mb')
new_range = dc.convertColorRange(dc.getRange(),old_unit)
dc.setRange(new_range)
pause()
image = getImage()
writeImage(output_name)
```



If you would like to change the contour interval from the default (4 mb in our example) to a new interval (let's go with 2 mb), simply do the following:

```
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
ctm = idv.getColorTableManager()
dc.setColorTable(ctm.getColorTable('White'))
old_unit = dc.getDisplayUnit()
dc.setDisplayUnitName('mb')
new_range = dc.convertColorRange(dc.getRange(),old_unit)
dc.setRange(new_range)
cinfo=dc.getContourInfo()
cinfo.setInterval(2.0)
dc.setContourInfo(cinfo)
pause()
image = getImage()
writeImage(output_name)
```



Note that the variable `cinfo` contains the various properties for the contours, including turing on/off labels, font type, dash style, alignment, etc. For more information on the methods that can be used to change the contour properties through the `cinfo` variable, see the JavaDoc for [`ControurInfo`].

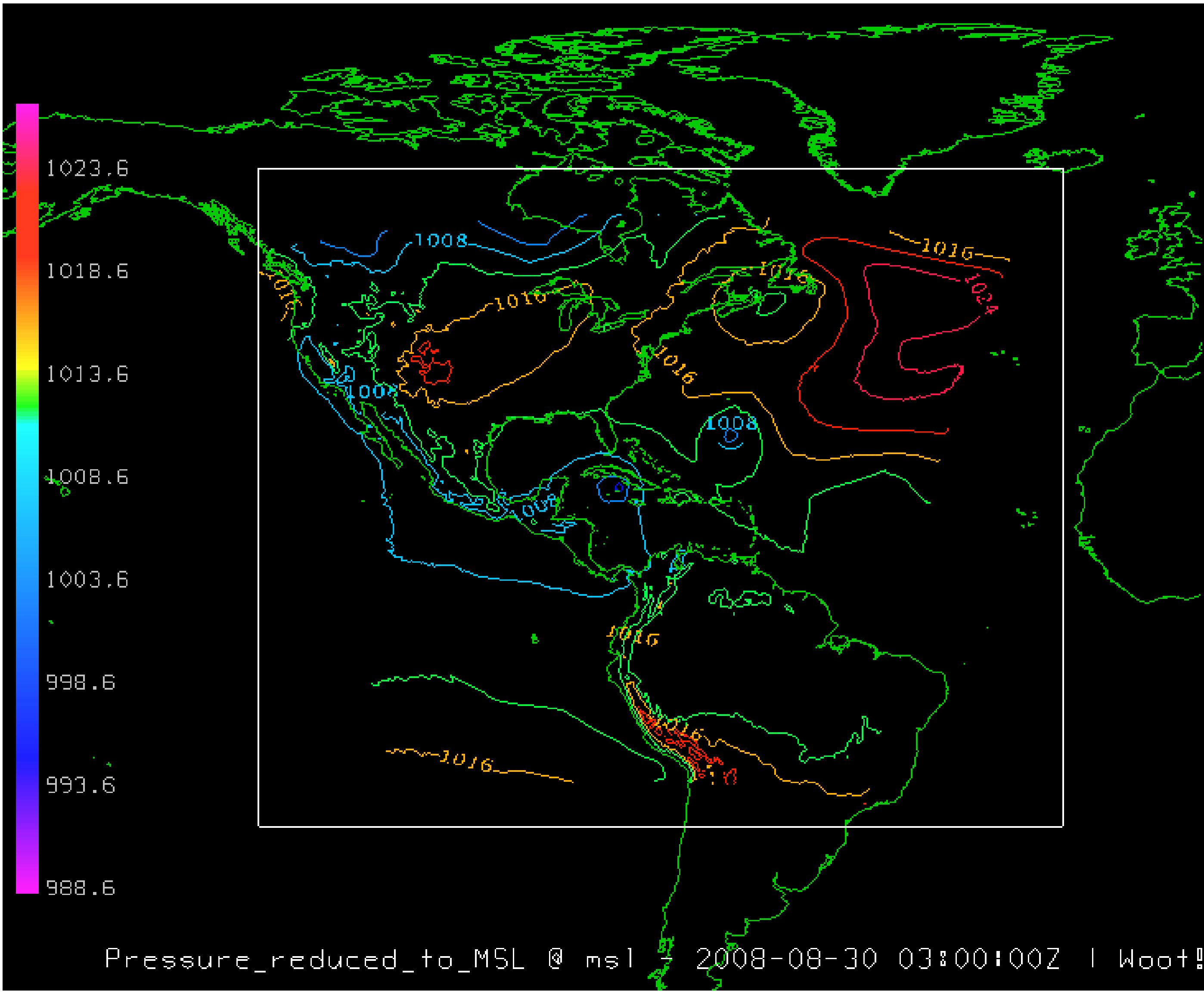
3.13.1.3 Labels and Color Scales

Labeling your images and adding a color scale.
The default display label for a variable can be very useful; however, sometimes it's not so great (as in our case, where the label says "Data - Contour Plan View - timestamp"). Changing the string that composes the label isn't much work, but changing the color of the text requires a bit more care. In order to change the color of the label text, we need to import the Java AWT class `Color` using `import java.awt.Color as Color`. The last bit of the import statement (`as Color`) allows us to refer to the java.awt.Color class by simply using Color (it reduces typing in the end if you have multiple calls to Color).

Built-in to the IDV display label is a macro sub-language. This will allow us to refer to information contained within the datafile, such as longname, shortname, and timestamp, through easy commands like %longname%, %shortname%, and %timestamp%. That said, sometimes the macro for things like %longname% and %shortname% return not-so-informative strings, like "Data" (which is what happens in our example). Not to worry, as you have complete control over the text in the label, as shown in the example below. Note that a list of display name macros can be found in source code for `DisplayControlbase` and are listed as variables named `MACRO_*`.

Also included in the example below is a demonstration of how a color scale can be added to your display. Note that all of the work is being done to a `ColorScaleInfo` object (which we have stored in the variable `colorScaleInfo`).

```
import java.awt.Color as Color
filename = './wrfprs_d01.060'
file_opener = 'file.grid'
ds = makeDataSource(filename, file_opener)
variable = 'Pressure_reduced_to_MSL @ msl'
display_type = 'planviewcontour'
image_dimensions = (800, 600)
output_name = 'contour2D.png'
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions[0], image_dimensions[1]))
pressure = getData(ds.getName(), variable)
dc = createDisplay(display_type, pressure)
ctm = idv.getColorTableManager()
dc.setColorTable(ctm.getColorTable('White'))
old_unit = dc.getDisplayUnit()
dc.setDisplayUnitName('mb')
new_range = dc.convertColorRange(dc.getRange(),old_unit)
dc.setRange(new_range)
dc.setDisplayListTemplate(variable+' - %timestamp% | Woot!')
dc.setDisplayListColor('Color.white')
colorScaleInfo = dc.getColorScaleInfo()
colorScaleInfo.setOrientation(colorScaleInfo.VERTICAL)
colorScaleInfo.setPlacement(colorScaleInfo.LEFT)
colorScaleInfo.setIsVisible(True)
pause()
image = getImage()
writeImage(output_name)
```

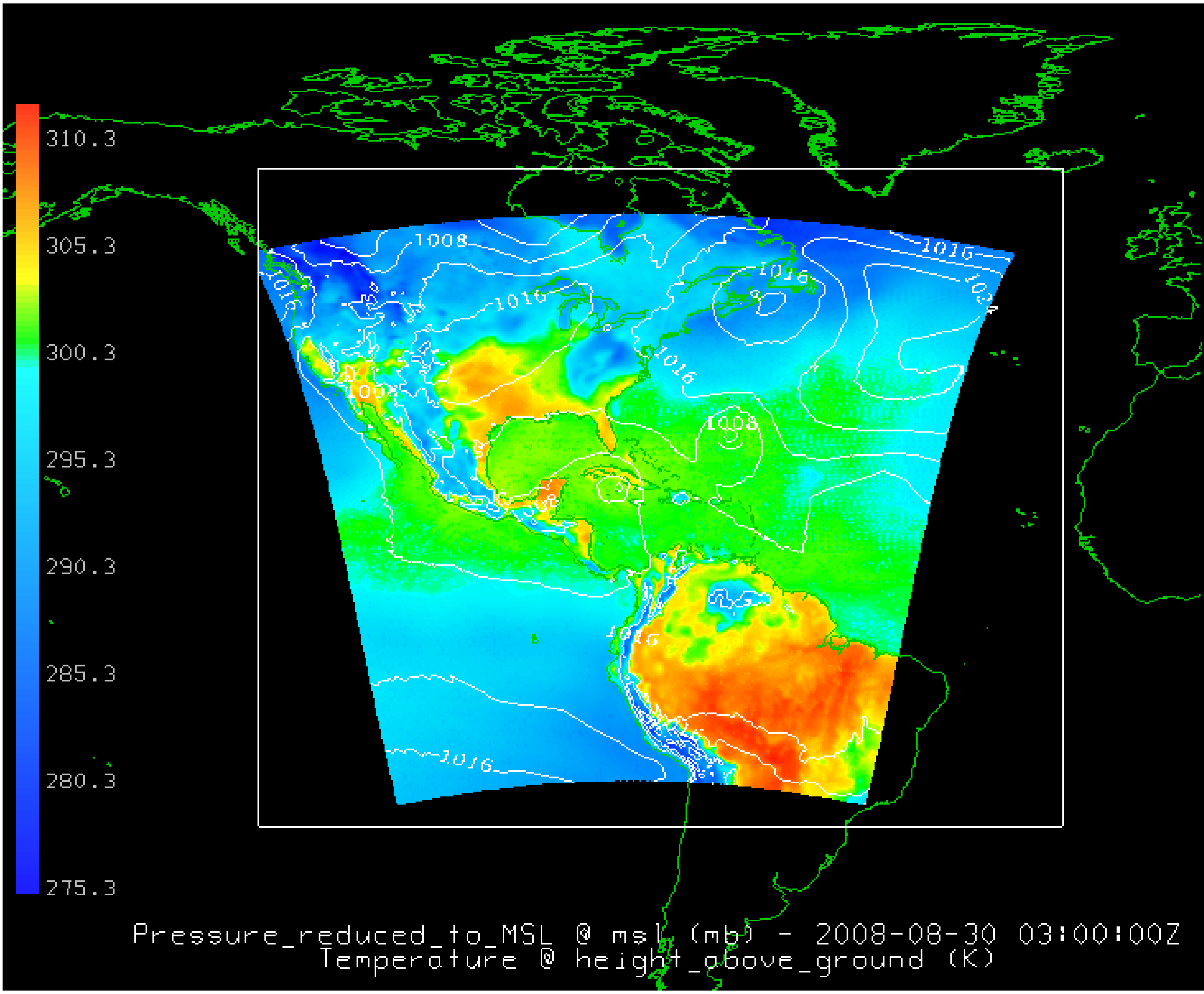

1023.6
1018.6
1013.6
1008.6
1003.6
998.6
993.6
988.6
1008
1016
1024
Pressure_reduced_to_MSL @ msl - 2008-08-30 03:00:00Z | Woot!

3.13.1.4 Combining Displays

Adding different displays to your WRF images.
Now that we have basic control over our display, let's add a *color-shaded plan view* to our contoured display. The following example code is based on the example from the previous section, but has been re-arranged for clarity. Note that we have given each `DisplayControl` object (`dc_contour` and `dc_shade`) a "ZPosition" to ensure that the contours are on top of the color shaded display.

```
import java.awt.Color as Color
#
# input options
#
filename = '../data/wrfprs_d01.057'
contour_variable = 'Pressure_reduced_to_MSL @ msl'
shade_variable = 'Temperature @ height_above_ground'
file_opener = 'file.grid'
#
# output options
#
display_contour = 'planviewcontour'
display_shade = 'planviewcolor'
image_dimensions = (800, 600)
output_name = 'twoDisplays.png'
#
# Generate the image
#
# setup the basic view of the IDV
#
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(image_dimensions\[0], image_dimensions\[1])
ctm = idv.getColorTableManager()
#
# get data
#
ds = makeDataSource(filename, file_opener)
contour_data = getData(ds.getName(), contour_variable)
shade_data = getData(ds.getName(), shade_variable)
#
# create colorshaded image
#
dc_shade = createDisplay(display_shade, shade_data)
# add label
label = shade_variable+' (%displayunit%)'
dc_shade.setDisplayListTemplate(label )
dc_shade.setDisplayListColor(Color.white)
dc_shade.setColorTable(ctm.getColorTable('PressureMSL'))
# add color scale
colorScaleInfo = dc_shade.getColorScaleInfo()
colorScaleInfo.setOrientation(colorScaleInfo.VERTICAL)
colorScaleInfo.setPlacement(colorScaleInfo.LEFT)
colorScaleInfo.setIsVisible(True)
dc_shade.setZPosition(0.)
pause()
#
# create contoured image
```

```
#
dc_contour = createDisplay(display_contour, contour_data)
dc_contour.setColorTable(ctm.getColorTable('White'))
# change unit
old_unit = dc_contour.getDisplayUnit()
dc_contour.setDisplayUnitName('mb')
new_range = dc_contour.convertColorRange(dc_contour.getRange(),old_unit)
dc_contour.setRange(new_range)
# create label
label = contour_variable+' (%displayunit%)'
dc_contour.setDisplayListTemplate(label+' - %timestamp%')
dc_contour.setDisplayListColor(Color.white)
dc_contour.displayableToFront()
dc_contour.setZPosition(1.)
# render and save image
pause()
image = getImage()
writeImage(output_name)
```


310.3
305.3
300.3
295.3
290.3
285.3
280.3
275.3
1008
1016
Pressure_reduced_to_MSL @ msl (mb) - 2008-08-30 03:00:00Z
Temperature @ height_above_ground (K)

4 Advanced Topics

This section will cover a variety of advanced topics within the IDV.

4.0 Installing the IDV

In this section, you will learn how to download and install the IDV from Unidata.

4.0.0 Downloading and Installing the IDV from Unidata

There are three ways you can download and install the IDV - installers, Java Web Start, and using individual JAR (Java Archive) files. Information on each of these methods can be found in the IDV User's Guide section on Downloading and Running the IDV. For this workshop, we will use the installer for Linux.

Downloading the IDV Installer

1. Open the Unidata homepage in a new window by clicking here.
2. To download the IDV, you must be registered as a Unidata user. If you have already registered, log in using your user name and password. If you have not registered or you have forgotten your password, log in as `idvuser@unidata.ucar.edu`. Use the password that the instructor gives you.
3. Once you are logged in, click on the **Downloads IDV** menu from the menu bar at the top of the page.
4. On the " IDV Downloads" page, click on the `Latest Release` link on that page.
5. From the "IDV Download and Installations" page, click on the `IDV 3.1 Installer for Linux/x86` link to start the download. You may be prompted for a location to save the file to - if so save it your home directory. Otherwise it will automatically be saved to your Downloads directory. Alternate Instructions for: (Windows) (Mac OS-X) (Solaris/SPARC) (Solaris/x86)
6. After the download is complete, close the extra browser window. You are now ready to install the IDV.

Footnotes:
Downloading via FTP
If you cannot download through the Unidata Downloads page, you can access the necessary file directly via FTP.

1. Download the file appropriate for your system from ftp://ftp.unidata.ucar.edu/pub/idv/IDV3.1.
2. Save the file to your home directory

Windows

From the "IDV Downloads" page, click on the `IDV 3.1 Installer for Windows` link to start the download. You will be prompted for a location to save the file to. Save it somewhere on your disk. Remember the directory where you save this file. You will need to know it when you run the installer.

Mac OS-X

From the "IDV Downloads" page, click on the `IDV 3.1 Installer for Mac OS-X` link to start the download. You will be prompted for a location to save the file to. Save it your desktop.

Solaris/SPARC

From the "IDV Downloads" page, click on the `IDV 3.1 Installer for Solaris/SPARC` link to start the download. You will be prompted for a location to save the file to. Save it your home directory.

Solaris/x86

From the "IDV Downloads" page, click on the `IDV 3.1 Installer for Solaris/x86` link to start the download. You will be prompted for a location to save the file to. Save it your home directory.

4.0.1 Using the Installer

Once you have downloaded the installer, you can simply execute the file to extract the program to your hard drive.

Linux/Solaris instructions

Note: If you downloaded a file other than `idv_3.1_linux-i386_installer.sh` or saved the installer as a different name, replace the filename listed in the steps below with the name of the file you downloaded.

1. Open a terminal window.
2. Change to your home directory or the directory to which you downloaded the installer (if you are not already there).
3. Run the shell file by typing:

   ```
   sh idv_3.1_linux-i386_installer.sh
   ```
4. Follow the prompts of the installer using the default selections.

Windows instructions

If you are running on Windows, see the Windows instructions

Footnotes:

Windows instructions

1. Use the windows system browser (e.g., "My Computer") to navigate to the directory where the file was downloaded.
2. Click on the file icon to start the installer. Follow the prompts of the installer.
3. Once the IDV is installed, you can start it by clicking on the icon or link created during installation.

4.1 Configuring the IDV

In this section we will cover how to configure various aspects of the IDV

4.1.0 Basic Preferences

In this section we will briefly cover how to change user preferences.

1. Select the **Edit Preferences** menu item to bring up the `User Preferences` window. This window consists of seven tabs.
2. Select the `General` tab.
    - The `Show Dashboard on Start` and `Show Help Tip Dialog` checkboxes allow you to control what windows are shown automatically.
    - The `When Opening a Bundle` section allows you to control the behavior of the IDV when a bundle is opened.
    - `Show Windows When They Are Created` controls if the display windows are shown or not when they are not docked in the Dashboard.
    - As we will discuss later, the `Resource Sitepath` is one way to specify where to look for application resources. The command line argument:
      `-sitepath`
      is saved as a user preference.
3. Go to the `Formats and Data` tab.
    - Try changing the `Date Format` and click `Apply`. Notice that the format of the times in the `Time Animation Widget` and `Display List` changes to the new format.
    - Change `Latitude/Longitude` format to a higher precision, e.g. "##0.00000" and click `Apply`. Move the cursor around in the view window and notice the precision of the values in the `Cursor Readout`.
    - The `Probe Format` is an html template for displaying data probe results. Try mousing over the text field to show a tooltip that describes the format.
    - The `Distance Unit` is used for controls and readouts that use distances.
    - `Sampling Mode` is used in the displays to determine how grids are sampled.
    - Tune performance and memory use with the `Caching`, `Max Image Size` and `Grid Cache Threshold`.
4. Select the `View` tab. These are preferences used in the map display to determine clipping, wireframe display, background color, etc.
    - Check the `Show Display List` option.
    - Uncheck the `Show Times in View` option.
    - Set the `Display List` font to URW Gothic, 14 point.
    - Change the `Display List` color to white.
    - Change the `Default Projection` to **US>CONUS**.
    - Check the `Show Logo in View` option
    - Press `Apply`.
5. Select the `Navigation` tab. This allows you to define different mouse and key bindings.
6. Select the `Toolbar` tab. This allows you top define what toolbar icons are shown on the toolbar.
7. Select the `System` tab. These are preferences used to change the system settings such as the amount of memory allocated to the IDV, data caching, threading and some Java 3D settings. In order for these changes to take effect, the user will have to restart the IDV.
    - `Memory` You can set the amount of memory allocated to the IDV as a percentage of the total system memory or a specific amount (in megabytes). On 32-bit systems, the maximum is 1.5 gigabytes (1500 MB).
    - `Thread Count` Taking advantage of mulitple core chip architectures, the user can set both the thread count for rendering and for data reading. For rendering, each time step for each

display is rendered separately, and 2 threads is optimal due to memory issues.

- Set the `Java 3D/Enable geometry by reference` If you have problems with spurious map lines, turn this off.

4.1.1 Display Settings

Display settings are a way to share properties among a group of display controls and to define configuration profiles when creating new displays.

1. Setup...
   - We are going to load in two different "NCEP NAM CONUS 80 km" model runs from the Catalog chooser under the IDV catalog. e.g., choose the 0Z and the 12Z.
   - Create a `Contour Plan View` of Temperature from each model.
   - In one of the displays change the Contour interval to be 3, change the color table to some other color table (e.g., VisAD), change the level to 500.
   - Add in some Location display (i.e., from the **Display Special Locations** menu).
   - Add in a Range and Bearing display.
2. Note we can't really have an apples to apples comparison of the two Contour displays because they differ in the interval, color table and level.
3. We could go and change these settings in one of the displays to match the other but imagine if you had, say, 4 displays to change. Lots of clicking.
4. Instead, go to the **Edit Display Settings** menu for one of the contour displays to bring up the Display Settings dialog.
   - The Display Settings dialog shows a list of properties from the selected display (highlighted in blue).
   - Note how the `Display Groups` under menu **Select** can be used to select different groups of displays.
   - Select both of the Plan View Contour displays in the Target Displays panel.
   - Change the highlighted display by right-clicking on another one in the list on the right and view the new Property values on the left.
   - Highlight the "500 hPA" Contour Display.
   - Select the `Color Table`, `Contour Settings` and `Level` properties on the left.
   - Press `Apply` to apply these selected properties to the selected displays.
5. We can also save the selected properties and use them when creating a new Display.
   - Select the properties you want to save (e.g., Level, Color Table, etc.)
   - Press **File Save Selected Properties**
   - In the dialog enter "Workshop" for the category and "500 MB Contours" for the name. Select `Applicable only to displays of type...`
   - Press `OK` and then close the Display Settings dialog.
6. Go to the Field Selector, clear the existing displays and select `Contour Plan View`.
7. In the `Settings` tab select the "500 MB Contours" and create the display.
8. Note the results.

4.1.2 Editing Parameter Display Defaults

In the previous section we covered how to create new color tables. We saw that for a `temperature` field there were certain default values for the color table and range that is used. In this section we will discuss how to change this default behavior using the `Parameter Defaults Editor`.

1. If you haven't done so already load in some example data and create a display of `temperature`.
    - Load in the `Sample RUC Data` data source.
    - Create a `Color-Shaded Plan View` display with a `temperature` parameter.
    - Note that the color table used is the *Temperature* color table and that the range is -90 to 45.
2. Open up the `Parameter Defaults Editor` by selecting the **Tools Parameter Defaults Editor** menu item.
3. There should be two tabs shown: `User Defaults` and `System Defaults`. The first tab shows the user editable parameter defaults. The second tab shows the system defaults.
4. Click on the `System Defaults` tab.
5. The first column shows the parameter name pattern that is used to match the parameter. More about patterns.
6. Change the default color table and range for `temperature` parameters.
    - Scroll down to the row that holds the `temp` parameter.
    - Right click on this row and select **Copy Row to Users Defaults**.
    - You should now see the `temp` row in the `User Defaults` tab and Parameter Defaults dialog shown.
    - Change the default color table by selecting the **Color Tables Workshop Comfort** menu item.
    - Change the range to -45 and 45.
    - Save your changes by clicking the `OK` button.
7. Test out this change.
    - Remove your displays by selecting the **Edit Remove All Displays** menu item.
    - Create a `Color-Shaded Plan View` display with a `temperature` parameter.
    - You should see the new *Comfort* color table and range being used.
8. What about setting defaults for new parameters?
    - Create a Color-Filled Contour Plan View using the `relative humidity at fixed height above ground`.
    - Bring up the Contour Properties Editor for the display and change the interval to 10 press `OK`.
    - Change the range to be 0-100
    - Change the Color Table to the **Basic Relative himidity**
    - From the display's **File** menu choose **File Save Save as Parameter Defaults**
    - The Parameter Defaults dialog is shown. Press `OK`.
    - Remove displays and recreate the above display. All should be better now.

Footnotes:
To load in the `Sample RUC Data` data source:

- Open the `Catalog Chooser` by either:
    - Choose the **Data Choose Data From a Catalog** menu item.
    - Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Sample Data RUC Grid** item.
- Select the `Add Source` button.

To create a `Color-Shaded Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature` field.
- In the `Displays` list, select `Color-Shaded Plan View` and press the `Create Display` button.

To create a `Color-Shaded Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature` field.
- In the `Displays` list, select `Color-Shaded Plan View` and press the `Create Display` button.

4.1.3 Editing Parameter Aliases

In this section we will cover how to define new aliases for your data.

1. If you haven't already, load in some sample data. Load in the `Sample RUC Data` data source.
2. Create a `Color-Shaded Plan View` display with a `temperature at fixed height above ground` parameter.
3. What's wrong with the display? Hint: we just got done defining default display characteristics for `temperature` fields.

Footnotes:
To load in the `Sample RUC Data` data source:

- Open the `Catalog Chooser` by either:
  - Choose the **Data Choose Data From a Catalog** menu item.
  - Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Sample Data RUC Grid** item.
- Select the `Add Source` button.

To create a `Color-Shaded Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature at fixed height above ground` field.
- In the `Displays` list, select `Color-Shaded Plan View` and press the `Create Display` button.

4.1.4 More on Editing Parameter Aliases

1. The answer is:
    - The color table is the *default* color table.
    - The range used is the range from the actual data.
2. The problem is that the IDV does not recognize that the name of the `temperature at fixed height above ground` field as a temperature.
3. To solve this problem, the IDV provides a data alias facility.
4. First we need to know the actual parameter name, not its description.
    - Select the **Help Details** menu item from the `Color-Shaded Plan View` display control that was just created.
    - You should see a detailed description of this display, including the data it uses:

```
...
Â T_fhg temperature at fixed height above ground from Geogrid data source
...
```

    - The parameter name is `T_fhg`.
5. Open up the `Alias Editor` from the **Tools Parameter Alias Editor** menu.
    - You should see 2 tabs: `User Aliases` and `System Aliases`
    - The first tab shows the user's editable aliases. The second tab shows the default system aliases.
    - Select the `System Aliases` tab.
    - Right click on the `TEMP` row and select **Edit Alias**.
    - You should now see this row in the `User Aliases` tab and the Data Alias dialog should be shown.
    - Scroll down to the bottom of the `Aliases` list, enter "T_fhg" and click the `OK` button.
    - Click the `Close` button in the `Alias Editor` window.
    - To edit an existing entry right click or double click on the parameter.
6. Let's see if this works.
    - Remove your displays by selecting the **Edit Remove All Displays** menu item.
    - Create a `Color-Shaded Plan View` display with a `temperature at fixed height above ground` parameter.
    - You should see the *Comfort* color table and the correct range being used now.

Footnotes:
To create a `Color-Shaded Plan View`:

- In the `Field Selector` tab of the `Dashboard` select the desired data source under the `Data Sources` list.
- In the "Fields" panel in the `Field Selector`, expand the "3D grid" tab.
- Select the `temperature at fixed height above ground` field.
- In the `Displays` list, select `Color-Shaded Plan View` and press the `Create Display` button.

4.1.5 Editing Parameter Groups

In this section we will cover how to define new groups of parameters.

1. A parameter group is used to define derived quantities described in a further section.
2. Open up the `Group Editor` from the **Tools Parameter Groups Editor** menu.
3. Kind of looks similar, doesn't it?
4. Try creating a new group called "TEMP_U_V" that is a TEMP,U and V

4.2 Diagnostic Functions: Formulas and Jython

The IDV provides methods to make computations using data in any IDV data source, and to display the results. You can use simple formulas or program code written in the Jython language.

4.2.0 Basics of IDV Formulas

Formulas in the IDV are one-line mathematical expressions to derive new values from data available.

The IDV comes with some Formulas. Here is an example of using one.

1. Using an IDV System Formula
   - Load in the `ETA 1998-06-29 00:00` data source.
   - Load in the `ETA 1998-06-29 12:00` data source, so we have two output times from the Eta model, 12 hours apart.
   - In the `Field Selector`, select `Formulas` in the `Data Sources` panel. Click on the `Miscellaneous` tag in the `Fields` panel
   - Right-click on `Simple difference a-b`, then click on `Edit Formula` in the popup menu. The `Formula Editor` appears. (For more see Formula Editor).
   - Click `Cancel` in the `Formula Editor`.
2. Requesting the Formula to be Computed and Displayed
   - In the `Displays` panel expand the Plan Views tab and click on `Contour Plan View`, then click on `Create Display`.
   - A `Field Selector` window appears that allows you to select which actual parameters from a data source or sources you want to use for each variable in the formula. You can select parameters from more than one data source for use in this single formula.
   - In the `Field Selector` window:
     For the "a" operand choose:
     `Eta 1998-06-29 12:00 -> 2D Grid -> mean sea level pressure (Eta model reduction)`

     For the "b" operand choose:
     `Eta 1998-06-29 00:00 -> 2D Grid -> mean sea level pressure (Eta model reduction)`
   - Click on `OK`.
   - Grids of twelve hour MSL pressure difference values are computed using the formula, and the result shown in the main view window.
   - Look at the contours and see if the values are reasonable. (What are the units?).

   You can make your own formulas. Your formulas are saved so you can build a library of your own derived quantities, using data you have.

3. Creating a new formula for wind speed from u and v wind components

   - Remove all displays.
   - From the main menu, click on **Edit->Formulas->Create Formula**. The `Formula Editor` window appears.
   - Enter the `Name windspeed` (one word - no space).
   - Enter the mathematical formula definition in the `Formula` entry field. The formula is
     `sqrt(u**2 + v**2)`
   - Open the `Advanced` panel.
   - Enter the `Description wind speed from u and v`.
   - For `Group` enter `Workshop`.
   - In the `Displays` section, click the `Use selected` radio button, then click the `All off` button.

- Expand the `Plan Views` category and check the `Contour Plan View` option. Also, expand the `3D Surface` category and check the `Isosurface` option.
- Click on `Add Formula`.
- A new item `wind speed from u and v` should appear in the `Field Selector` window's `Fields` panel, under `Workshop`.

4. Sub-setting times in the data source
   - To save computation time, set the `12:00 Z Eta` data source to use the first three times, with the `Field Selector` window. (see Selecting Times).
5. Requesting the new Formula to be Computed and Displayed
   - Click the `wind speed from u and v` item in the `Fields` panel under the `Workshop` group.
   - In the `Displays` panel click on `Contour Plan View`, then click on `Create Display`.
   - In the `Field Selector` window:
     For the "u" operand choose:
     `Eta 1998-06-29 12:00 -> 3D Grid -> u component of wind`

     For the "v" operand choose:
     `Eta 1998-06-29 12:00 -> 3D Grid -> v component of wind`
   - Click on `OK`.
   - Grids of wind speed are computed using the formula, and the display is made. All three data times selected for the data source are used.
   - Look at the contours and see if the speed values are reasonable. Go on to the next step to examine the wind speed in more detail.
6. Compare to the IDV's derived wind speed
   - Use the `Levels` selector to change the wind display level to 250 hPa.
   - From the `Eta 1998-06-29 12:00` choose the parameter `3D Grid -> Derived -> Windspeed (from Gridrelative_u and Gridrelative_v)`

     Make a Contour Plan View of this wind speed automatically derived by the IDV.
   - Change its level to 250 hPa.
   - Toggle between the two displays to check for differences.
7. Extra - Calculate a difference between the system derived wind speed and the wind speed from your local formula using the simple difference formula. What would you expect the result to be? Answer

Notes on formulas

The `Description` is listed in the `Field Selector`'s `Fields` panel. If you leave the `Description` field empty, the name of the formula will be used as the description.

`Group` is a way of categorizing your formulas and is optional. If the `Group` does not already exist, it is added when you save the new formula. If the `Group` already exists, the new formula will appear under that group's tab in the `Field Selector`'s `Fields` panel. If you leave the `Group` empty, the formula appears in the list directly.

The variable names in the formula definition, such as dpt, are dummy names, and in principle can be anything, such as "a" or "var2". It is best to use variable names that suggest the parameter data they represent so that later in the parameter selection step you remember what the variables should represent. Use names like Temp500m, RelHum_surf, absvort, sst_jan, or whatever makes sense to you. If you use a variable names like

V1 and V2, then later when the formula requests which real parameter name goes with which dummy variable name you may be puzzled.

Once you have defined a formula, it is saved and will appear in future runs of your IDV. It is saved in your personal copy of the derived.xml file, which usually is in the file `~/.unidata/idv/DefaultIdv/derived.xml` on UNIX systems.

Holding the mouse pointer stationary over the formula name in the `Field Selector`'s `Fields` panel will cause a tooltip box to appear showing the formula name and the mathematical formula.

Formulas are preserved by the IDV. Next time you start the IDV you will see formulas you created before. You can build up a library of your own formulas.

To remove a formula, click on `Remove formula` in the `Field Selector`'s formula pull down menu. This does not remove a display of calculations made with a formula, it removes the formula itself from the `Field Selector` window. Usually you leave formulas in place until you are sure you will not use them again.

Footnotes:
To load in the `ETA 1998-06-29 00:00` data source:

- Open the `Catalog Chooser` by either:
  - Choose the **Data Choose Data From a Catalog** menu item.
  - Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Case Studies Data for Comet Case Study 039 NCEP Model Data ETA 1998-06-29 00:00 GMT** item.
- Select the `Add Source` button.

- Use the **File Save Data in Cache** menu in the control for the `Windspeed from u and v` display to save the data to the cache. When the dialog pops up, enter `windspeed (formula)`.
- Use the **File Save Data in Cache** menu in the control for the `Windspeed (from GridRelative_u & GridRelative_v)` display to save the data to the cache. When the dialog pops up, enter `windspeed (system)`.
- In the `Field Selector` window, click on the `Formulas` data source and select the `System -> Simple difference a-b` field and the `Contour Plan View` display, then click the `Create Display` button.
- From the `Field Selector` window that pops up, select one of the cached windspeeds for a and the other for b, then click `OK`
- The fields will be differenced and nothing will display because they have the same values.

Formula Editor

The `Name` is a short name used for the legend labels on displays.

Enter the mathematical formula definition after `Formula`.

IDV uses the Jython language for formulas. Jython supports such common mathematical operators as +, -, *, /, and ** (for exponentiation). Rules of precedence apply, for example, 1.5*T*W**2 is the same as (1.5*T) * (W**2).

To perform common mathematical operations on two gridded VisAD objects a and b, we recommend user to use the corresponding grid diagnostic formulas: add(a,b), sub(a,b), mul(a,b), and quo(a,b). This can be less problematic. The simple different formula a-b in this exercise is calling sub(a, b).

The variable names in the formula definition are arbitrary place holders. Do not use names with spaces such as "wind speed." Start each variable name with a letter.

The advanced section allows you to define a lengthier description, a group and what displays to use.

The `Description` appears in the `Field Selector` window. If nothing is entered here, the formula name (above) is reused for a description.

A `Group` will create a new (and optional) group in the `Field Selector` window under `Formulas`. It can be any normal text including spaces, numbers, and punctuation characters, such as "moisture parameters" or "two level profiler difference, English units." You do not need to enter a group.

In the `Displays` section, either accept the default `Use all`, or select the `Use selected` button and click off the displays that this formula's result will not support. For example, a field of wind vectors does not need to have a display choice of Isosurface.

4.2.1 Jython Methods

You have seen how to compute wind speed from u and v wind components, using the Formulas facility. IDV formulas are actually code in the Jython language. If your calculations require more than a one-line mathematical expression you can create Jython methods to make calculations then you can call these methods in a formula. Here is an example of computing wind speed with a four-line method.

1. Invoking the Jython Library and Editor
    - Click on the main menu item **Edit Formulas Jython Library** to bring up the `Jython Libraries` Editor

   The Jython Library window has two or more panes selected by the tabs at the top. Only the "User's Library&quot is editable; the others came from the site path and system defaults.

   How to Edit in the Jython Library and for more info see How to write a Jython Method and Jython Methods and the IDV

2. Defining your own Jython Method
    - In the `Jython libraries` Editor, click on the tab `User's Library`
    - Create a new Jython method called windSpeed(u,v). Jython methods have a first line beginning with "def" and ending with the colon ":". The code block composing the method must be indented at least one space. Indentation must be the same for all line in a code block, except continued comment lines which are set off with triple quotes. Your code block will be something like:

    ```
    def  windSpeed(u,v):
         """  compute wind speed from u and v
              wind components """
         usqd= u*u
         vsqd= v*v
         ws = sqrt(usqd + vsqd)
         return ws
    ```
    - Click on `Save` to check for correct Jython syntax and save the method; you will get an error message if the syntax is not correct. If it is complete, your method is saved.
    - Use the **File Close** menu to exit the editor.
3. Use the Method in a Formula
    - Leave up your previous display made with formulas, and the same data source.
    - From the main menu, click on **Edit Formulas Create Formula**. The `Formula Editor` window appears.
    - Enter the `Name` `jythonWS`.
    - Enter the method for the formula definition in the `Formula` entry field, simply `windSpeed(u,v)`.
    - Open the `Advanced` panel by clicking on the down arrow.
    - Enter the `Description` "jython wind speed".
    - For `Group` select `Workshop` from the drop down list.
    - In the `Displays` section, click on `Use selected` and then the `All off` radio button. Then select `Contour Plan View` and `Contour Vertical Cross Section` in the list.
    - Click on `Add Formula`.
    - An item `jython wind speed` should appear in the `Field Selector` window's `Fields` panel, under the `Workshop` category.

- ♦ Use the formula to make a contour display of wind speed using the same u and v wind components from the previous exercise.
- ♦ Check to see if the results appear reasonable.

Footnotes:
The user library pane is an editor window you can type in. It uses simple mouse actions to move the insertion point and mark areas for cut and paste. Drag the mouse to highlight; use the delete or backspace key to remove highlighted text, and middle mouse button to paste.

How to write a Jython Method

A Jython method starts with the word **def**. The first line includes the method name and argument list, and **ends with a colon**. Method names are, by convention, descriptive with each first letter of a word in upper case, except the first letter in the method name is always lower case. Using this convention means others can use your code with less trouble. Do not omit the final colon on this line.

Method documentation, which is optional but highly recommended, is enclosed in sets of triple quotes beginning in the first line after "def." with first line setting the indent level for the method. Regular comment lines begin with #, and can be located anywhere.

Code blocks correspond to the same level of indentation, such as the code block in these methods. There is no extra end-of-line character. There is no declaration of new variables, you simply use variables as needed and the type of data is deduced by Jython from the context.

Indentation must be the same in Jython code blocks or it will not run, or fail to run correctly. How many spaces you use for a code block indent is up to you, but must be one or more spaces.

Jython syntax is checked when you save the file. If there is an error, a pop-up window will appear with the message **An exception has occurred writing in Jython library**. The actual error will also be printed in the console window running the IDV. Correct the error and try to save again.

Common Jython errors include irregular indentation, and omitting the colon from the end of for, if, and def statements.

Note that you can cut and paste from the Jython Library Editor to the formula editor. Keeping the Jython Library Editor open while creating formulas is handy.

Jython Methods and the IDV

In any Jython method in the IDV the operands and results can be used for single values, multidimensional arrays, or any other data object supported by the IDV and VisAD, depending on the type of the incoming data.

The example simple method is so short it would have been easier to simply type it as one line formula. But any method can be called by other methods in the library, or by formulas. There is no limit on how long your methods can be.

Your personal Jython library is stored in a file named `default.py` in your home directory or in a directory under the home directory, usually in `~/.unidata/idv/DefaultIdv/default.py` on UNIX systems. This one file keeps all your Jython methods.

The Jython Library Editor automatically imports the methods in visad.python.JPythonMethods. You can use them simply by typing the method names in your methods.

Check out the methods in the system Jython library to see if they are of use to you. These methods also show some examples of calling IDV internal code.

4.2.2 Doing More With Formulas

Formulas can be used to do simple arithmetic such as the simple difference, or they can be used to call Jython procedures as in the previous excercise. They can also be used to call Java methods and nest Jython procedures. In this exercise, we will show how to create more complex formulas from the set of diagnostic functions in the IDV.

First review Basics of IDV Formulas.

1. Calculating the ageostrophic wind
   - Clear all displays and data.
   - Load in the `ETA 1998-06-29 00:00` data source.
   - Create the following displays:
     - 500 mb contours of geopotential height
     - 500 mb flow vectors of grid relative winds. Set the skip interval to 2 and the range to be -30 to 30. Save the display settings as `Workshop->Flow fields`.
     - 500 mb flow vectors of the geostrophic wind. Use the display settings saved in the previous step.
   - The ageostropic wind is the difference between the geostropic wind and the actual wind.
     `AGEO = OBS - GEO`
   - We could simply create a formula that is OBS - GEO and select the flow vector field and the geostropic wind. For this exercise, we will look at how we can do this from the basic fields, on the fly.
   - Open the Jython Library and look at the **System Grid Diagnostics** library. This is a set of diagnostic functions that can be applied to grids. The names closely match the set of grid diagnostics in GEMPAK. Look at the page in the Jython Libraty section of the User's Guide.
   - From the Formulas Data Source, select the **Grids Define a grid diagnostic** formula.
   - Select the **Flow Displays Vector Plan View**, the `Workshop->Flow fields` display setting and create the display.
   - You will be prompted for the formula. In the dialog, type in:
     `vsub(vecr(u,v),geo(z))`

     This will take the vector difference (**vsub**) between the vector (**vecr**) defined by **u** and **v** components and the geostrophic wind (**geo**) calculated from the geopotential height field (**z**).
   - Click the `OK` button.
   - For the fields, select the u and v wind components and the geopotential height for u,v and z respectively. Select the 500 hPa level for each of the grids.
   - Zoom in to see the detail of the plots.
2. Calculating the magnitude of the ageostrophic wind
   - The magnitude (speed) of the ageostrophic wind can be calulated using the mag function. To save some steps, we'll save the ageostropic wind field in the cache and use that.
   - Use the ageostrophic vector control's **File Save Save Data in Cache** menu to save the data in the cache. You can leave the name as is or change it to ageo.
   - Select the **Grid Create a grid diagnostic** formula and the Contour Plan View Display.
   - In the dialog, type:
     `mag(v)`
   - For the field, select the cached data.

Footnotes:

To load in the `ETA 1998-06-29 00:00` data source:

- Open the `Catalog Chooser` by either:
    - ♦ Choose the **Data Choose Data From a Catalog** menu item.
    - ♦ Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Case Studies Data for Comet Case Study 039 NCEP Model Data ETA 1998-06-29 00:00 GMT** item.
- Select the `Add Source` button.

4.2.3 Doing More with Jython

(Courtesy of Dr. William Fingerhut, Lyndon State College, VT. For full details, see http://apollo.lsc.vsc.edu/~fingerhutb/UserWshop2004/W2004_IDV_LabF.html

A vertical cross section of temperature does not show as much structure as possible because the vertical temperature gradient is much larger than the horizontal temperature gradient. One way to solve this problem is to subtract the U. S. Standard Atmosphere from the temperature field, and create a vertical cross section of temperature anomaly.

This exercise will show how Jython can be used in the IDV for complex calculations like this.

1. Load in the `Sample RUC Data` data source.
2. Create a `Contour Vertical Cross Section` of the `3D grid->temperature` field
3. Move the cross section line in the main display to a North/South orientation by dragging the endpoints.
4. View the Jython procedure for calculating temperature anomaly
   - Bring up the Jython library editor with the **Edit Formulas Jython Library** menu item.
   - Click on the `..workshop/default.py` tab which is the Jython library loaded from the plugin. You should see:

     ```
     def tempAnom(t):
         """ temperature anomaly from U.S. Standard Atmosphere """
         # get pressure levels of temperature grids
         p=extractPressureFromNWPGrid(t)
         # calculate temperature for a constant lapse rate (6.5 C/km) atmosphere
         tstd=288.15*(p/1013.25)**(287.05*.0065/9.806)
         # change temperature in stratosphere to isothermal (216.65 K)
         for i in range(len(p)):
           if p[i] < 225.0:
             tstd[i]=216.65
         # change the units to K
         tstd = newUnit(tstd,"tstd","K")
         # calculate the temperature anomaly
         tanom=t-tstd
         return tanom
     ```

     If it is not there, paste this formula into your locally editable library and click the `Save` button.
5. Create a Formula to call the tempAnom procedure
   - Select the **Edit Formulas Create Formula** menu item.
   - In the Formula Editor enter the following information:
     - `Name`: anomaly of temperature
     - `Formula`: tempAnom(temperature)
     - `Group`: Workshop
   - Click the `Add Formula` button.
6. Create the Display
   - Select the `Formulas` data source in the `Field Selector`.
   - Choose the `anomaly of temperature` formula under the Workshop tab in the `Fields` panel.
   - In the `Displays` pane select `Contour Vertical Cross Section`

- ♦ Click the `Create Display` button
- ♦ When prompted for a temperature field, select it from the sample RUC data and click the `OK` button

7. Share the cross section locations
   - ♦ In the **Edit Sharing** menu of each display control, check the **Sharing On** item.
   - ♦ Position the control windows and the main display so you can view all three at once.
   - ♦ Move the north/south line for the temperature cross section slightly to align the two transects.
   - ♦ Compare the two displays. A vertical cross section of temperature anomaly can discern:
     - ◊ the vertical level of temperature anomalies,
     - ◊ the horizontal location of large horizontal temperature gradients, and
     - ◊ the anomaly reversal across the tropopause.
8. Try this exercise with real-time model data.

Footnotes:
To load in the `Sample RUC Data` data source:

- Open the `Catalog Chooser` by either:
  - ♦ Choose the **Data Choose Data From a Catalog** menu item.
  - ♦ Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Sample Data RUC Grid** item.
- Select the `Add Source` button.

4.2.4 Jython and VisAD

How the IDV uses the Jython language and the VisAD software library.

Jython and the IDV

Jython is the Python computer language, implemented in Java. The Jython user sees regular Python and no Java. You do not need to know Java to use Jython. Python is a new language in which simple things are usually simple to do. IDV formulas use syntax and methods from Jython. Jython (Python) has a simple syntax, but this is a powerful computer language that can do most everything C++ and Java can do.

Jython has an easy and intuitive syntax. Jython is as easy to write as FORTRAN, and easier than C or C++. If you have existing FORTRAN code, conversion should be straightforward, with line by line conversions in most cases. In IDV formulas you can use any normal built-in Python (Jython) operator, such as +, -, / *, and ** for exponentiation, or numeric functions, such as abs, round, pow, or sqrt. Jython is also used in defining your own methods, described in this section.

The IDV uses Jython for formulas for three reasons. One is that Jython can call Java code, so there can be full integration between Jython methods and internal code in the IDV (you can see examples in the IDV Jython system library). Another reason is that the IDV uses VisAD software to store and manage data, and VisAD software is written in Java. VisAD also can be called directly by Jython. To work with data in the IDV you use VisAD. And finally, because Jython is easy as most any computer language.

You do not need to study Jython/Python in detail to use VisAD formulas. If you do wish to learn more about Jython and Python, we recommend these books

The Quick Python Book by Daryl Harms, Manning, 2000,
Jython for Java Programmers by Robert Bill, New Riders, 2002,
Python Essential Reference 2nd edition by David Beazley, New Riders, 2001,
Python Developers Handbook by Andre Lessa, Sams, 2001. and these web sites:
http://www.python.org,
http://www.jython.org,
Using Python with VisAD (http://www.ssec.wisc.edu/~tomw/visadtutor/).

VisAD and the IDV

The IDV also provides and supports VisAD, a Java library to provide display and analysis code for multi-dimensional data. All data objects used by the IDV are actually, invisibly to the user, stored in VisAD data objects, which carry with them information about units, errors, navigation coordinate systems and transformations, and other metadata.

VisAD includes a package of code, visad.python.JPythonMethods, to allow Jython calls to operate on VisAD data objects, such as 3D grids. That is why in the IDV you can write a simple formula such as (TC*1.8 + 32.0) - DewptF and it will automatically apply the calculation to every element in two large 3D grids referred to as TC and DewptF. No "do loops" over all grid points are needed; and you don't even need to know anything about how the grid is defined, or the data format, or the the grids's navigation on the Earth. The same "IDV Formula" can be applied, without alteration, to a single value, to data from a vertical sounding or a balloon track, or 2D grids or other 3D grids. All you need is one line of code. You can even mix data sources in one calculation: a temperature variable "TC" can come from Eta forecasts and another "DewptF" from GFS, mapped differently on the globe. All you need enter is a formula such as "(TC*1.8 + 32.0) - DewptF" and the rest is automatic.

You may use the IDV and its computation features without knowing anything about VisAD. However, to do more, you will want to see the documents for visad.python.JPythonMethods in the VisAD API Documentation. In your web browser, in the upper left box in this page, scroll down to visad.python and click on it. In the lower left box click on JPythonMethods. For complete information about VisAD, see the VisAD web site.

4.2.5 Derived Data and Formulas

The IDV uses the Formula infrastructure to also automatically create derived products from loaded data sources. These derived products show up in the list of Fields under a "Derived" tab.

1. Let's look at the Horizontal Advection formula
   - From the **Edit Formulas Edit Formulas Grids** menu, select the `Horizontal Advection (from %N1% & %N2%)` item.
   - The formula looks like:

     ```
     DerivedGridFactory.createHorizontalAdvection(scalar_parameter, D1[label=U Co
     ```

     The `D1 and D2` fields can be automatically bound to specific parameters and `scalar_parameter` is a field that the user will be prompted for.
   - In the `Description`, the `%N1% and %N2%` fields correspond to the names of the bound parameters in the order listed in the formula (not the number after the **D**]
   - Select the `Derived` tab.
   - This formula is used as both an end user formula and as a way to automatically create derived quantities. It uses predefined parameter group called `u_and_v` to find variables in a dataset that can be used to fill in for the `D1 and D2` parameters.
   - Click the `Cancel` button to close the window.
   - Select the `Formulas` Data Source.
   - Select the `System->Horizontal Advection (from %N1% & %N2%)` field, select `Contour Plan View` display and click `Create Display`. Notice how the formula relates to the dialog that pops up.
   - Click the `Cancel` button to close the window.
2. Display Temperature Advection from the derived formula list
   - Load in the `ETA 1998-06-29 00:00` data source.
   - Expand the **3D grid Derived** tab in the `Fields` panel of the `Field Selector` and select the `Horizontal Advection (from GridRelative_u & GridRelative_v)` field. Hold the mouse over the label to look at the tooltip which shows the formula for this field. Notice that the `%N1% and %N2%` parameters in the formula description have been filled in.
   - Select the `Contour Plan View` display and click `Create Display`.
   - When prompted for the `scalar parameter`, select the `3D grid->temperature` field and click `OK`.
3. Now, let's create a system formula for Temperature advection that will show up when a grid is loaded.
   - Select the `Formulas` Data Source.
   - Right click on the `System->Horizontal Advection (from %N1% & %N2%)` field and select **Copy Formula**.
     - Change the `Name` to `tadv`.
     - Change the `Formula` to `DerivedGridFactory.createHorizontalAdvection(D1, D2, D3)`.
     - Change the `Description` to `Temperature Advection (from %N1%, %N2% & %N3%)`.
     - Change the `Group` to `Custom Derived`.
     - Select the `Derived` tab.
     - Clear out the `u_and_v` under Parameter Groups.

        - ◊ Look at the list of available groups. We could use the group created in the prior exercise using the `Parameter Groups Editor` but for now we will enter the parameter names to match on directly.
        - ◊ In the `Parameters` input area, right click and select the Temperature, U component and V components (one at a time) from the **Aliases Group #1** menu available from the button or type `TEMP, UREL, VREL`.
        - ◊ Click the `Change Formula` button. The `Formulas` data source now has a `My Derived` category in the `Fields` panel.
    - ♦ Right click on the `NCEP Model Data` data source and select **Reload Data** to refresh the list of parameters. Derived data selections are only created when the Data Source is loaded or reloaded.
    - ♦ Select the `My Derived->Temperature Advection (from T, GridRelative_u & GridRelative_v)` parameter and the `Contour Plan View` display and then click the `Create Display` button.
    - ♦ Compare the two displays.
    - ♦ Remove the displays, but not the data.
    - ♦ From the `Formulas` data source, right click on the `My Derived->Temperature Advection (from %N1%, %N2% & %N3%)` field and select **Edit Formula**.
    - ♦ Remove the `Group` for the formula and click `Change Formula`.
    - ♦ Reload the `NCEP Model Data ETA` datasource.
    - ♦ Where did the derived formula go this time?
    - ♦ Use the `Files` chooser to load in the `sample_eta_grid.nc` file.
    - ♦ Does this have a temperature advection field?

4. Let's just look at the `derived.xml` file in:

   /home/idv/.unidata/idv/DefaultIdv

   to see what the formula looks like.

Footnotes:
To load in the `ETA 1998-06-29 00:00` data source:

- Open the `Catalog Chooser` by either:
    - ♦ Choose the **Data Choose Data From a Catalog** menu item.
    - ♦ Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Case Studies Data for Comet Case Study 039 NCEP Model Data ETA 1998-06-29 00:00 GMT** item.
- Select the `Add Source` button.

4.3 Scripting with IDV

The IDV offers a couple of different scripting environments. The IDV Scripting Language (ISL) is an XML based language for scripting the IDV. The IDV can also be scripted via Jython programming language. You can load bundles, create images and movies,manipulate files, etc. User Guide.

4.3.0 ISL Introduction

Create a simple "Hello World" ISL file.

1. First of all bring up a terminal window and create a sub-directory: `/home/idv/isl`
2. Change directory to the `/home/idv/isl` directory.
3. Copy the following to your `isl` directory. Right click on link and do 'Save Link As'

```
<isl debug="true">
     <!-- This is an XML comment -->
     <echo message="Hello world"/>
</isl>
```

   basic1.isl
4. From the terminal run the IDV with the basic1.isl as the command line argument:

```
idv /home/idv/isl/basic1.isl
```

5. Lets loop this now by adding a ***group*** tag. Copy this file:

```
<isl debug="true">
     <echo message="Here is my loop:"/>
     <group loop="5" sleep="1">
         <echo message="Hello world"/>
     </group>
     <echo message="Done"/>
</isl>
```

   basic2.isl
6. Let's define some properties with the ***property*** tag. Properties are referenced as *${propertyname}*.

```
<isl debug="true">
     <property name="message1" value="This is  property message1"/>
     <property name="message2" value="This is  property message2"/>
     <echo message="Here is my loop:"/>
     <group loop="5" sleep="1">
         <echo message="Here is message 1: ${message1}"/>
         <echo message="Here is message 2: ${message2}"/>
         <echo message="There is a built-in loopindex property: ${loopindex}"/>
     </group>


     <echo message="

There are also built-in time properties"/>
     <echo message="The islpath points to where the isl is: ${islpath}"/>
     <echo message="There are date/time properties, e.g.: ${yyyy}_${MM}_${dd} ${H}:${mm}:$


</isl>
```

   basic3.isl

4.3.1 Generating Images

Load in a bundle and generate a screen shot.

1. First of all save this isl.xidv file to your isl directory.
2. Run this file:

   ```
   <isl debug="true" offscreen="false">
        <bundle file="isl.xidv"/>
   </isl>
   ```

   image1.isl
3. Now, change directory to the home directory (cd) and run:

   ```
   idv /home/idv/isl/image1.isl
   ```
4. What happened?
5. The bundle reference in the isl was relative to where you run the idv from. We'll use the *islpath* property:

   ```
   <isl debug="true" offscreen="false">
        <bundle file="${islpath}/isl.xidv"/>
   </isl>
   ```

   image2.isl
6. Note, no images were created. Only the bundle was loaded. To create an image we'll use the *image* tag:

   ```
   <isl debug="true" offscreen="false">
        <bundle file="${islpath}/isl.xidv"/>

   <!-- Wait until everything is loaded -->
        <pause/>

   <!-- Take a screen shot -->
        <image file="${islpath}/isl.png"/>
   </isl>
   ```

   image3.isl
7. The *movie* tag lets us create a time-loop movie or animated gif.

   ```
   <isl debug="true" offscreen="false">
        <bundle file="${islpath}/isl.xidv"/>

        <pause/>

   <!-- Generate an quicktime movie -->
        <movie file="${islpath}/isl.mov"/>

   <!-- Generate an animate gif -->
        <movie file="${islpath}/isl.gif"/>
   </isl>
   ```

   image4.isl

8. The IDV gives you an easy way to save off a basic ISL file. Run the IDV with the above bundles. Select **File Save As...** and choose the `/home/idv/isl` directory and enter `image.isl`.
9. At the ISL Properties dialog:
    1. Turn off Offscreen.
    2. Turn off Image
    3. Turn on Movie and enter `image.kmz`.
    4. Press OK. This saves a bundle (image.xidv) and an ISL file.
10. The ISL file should look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<isl debug="true" loop="1" offscreen="false" sleep="3600">
  <bundle clear="true" file="${islpath}/image.xidv" wait="true"/>
  <movie file="${islpath}/image.kmz"/>
</isl>
```

    image.isl
11. If we enter a `kmz` suffix for the movie the IDV will generate a Google Earth kmz file of the animation loop.
12. Now, we'll show you an easier way to do all of this.
    - Bring up the IDV again and load the above bundle.
    - Now, do a **File Save As** and this time call it `idv.isl`.
    - Select `Generate a Movie` and press `OK`.
    - Lets look at this file.
    - Now, save the bundle as idv2.isl and this time select `Include Bundle Inline` and press `OK`.
    - Look at the result.

4.3.2 Manipulating Images

The ISL supports a variety of image manipulations.

1. We can manipulate the image in different ways:

```
<isl debug="true" offscreen="false">
     <bundle file="${islpath}/isl.xidv"/>
     <pause/>

     <!-- Make an image -->
     <image file="${islpath}/clipped.png">
          <!-- Clip it at the lat/lon box -->
          <clip north="60" south="20" east="-60" west="-140"/>

          <!-- resize it -->
          <resize width="400"/>

          <!-- Make a thumbnail -->
          <thumbnail file="${islpath}/clipped_thumb.png" width="25%"/>
     </image>

     <!-- Make another image -->
     <image file="${islpath}/matted.png">
          <!-- add a logo overlay -->
          <overlay image="http://www.unidata.ucar.edu/software/idv/logo.gif"
                   place="LL,10,-10"
                   anchor="LL"/>

          <!-- Matte the image -->
          <matte background="red" bottom="100" top="100"/>

          <!-- Overlay some text -->
          <overlay text="Workshop Example" place="LM,0,-10" anchor="LM" color="blue" fonts

     </image>
</isl>
```

   image5.isl
2. Note: everything we can do with the image tag we can also do with the movie tag:

```
<isl debug="true" offscreen="false">
     <bundle file="${islpath}/isl.xidv"/>
     <pause/>

     <movie file="${islpath}/overlay.mov">
          <!-- add a logo overlay in the upper left -->
          <overlay image="http://www.unidata.ucar.edu/software/idv/logo.gif"
                   place="ul,10,10"
                   anchor="ul"/>
     </movie>
</isl>
```

   image6.isl

4.3.3 ISL Exercises

Come up with an interesting ISL file. Here are some ideas:

1. Do some different image manipulations. Add in your own logo or text annotation.
2. Create a bundle with real time radar. Loop 10 times, sleeping 10 minutes each loop. Call *reload* to reload the data. Capture an image. Use the date/time properties in your filename.
3. Try to create a procedure that loads a bundle, and writes an image. Call it with a set of bundles from a *foreach* loop tag.

4.3.4 Interactive Scripting with the IDV and the Jython Shell

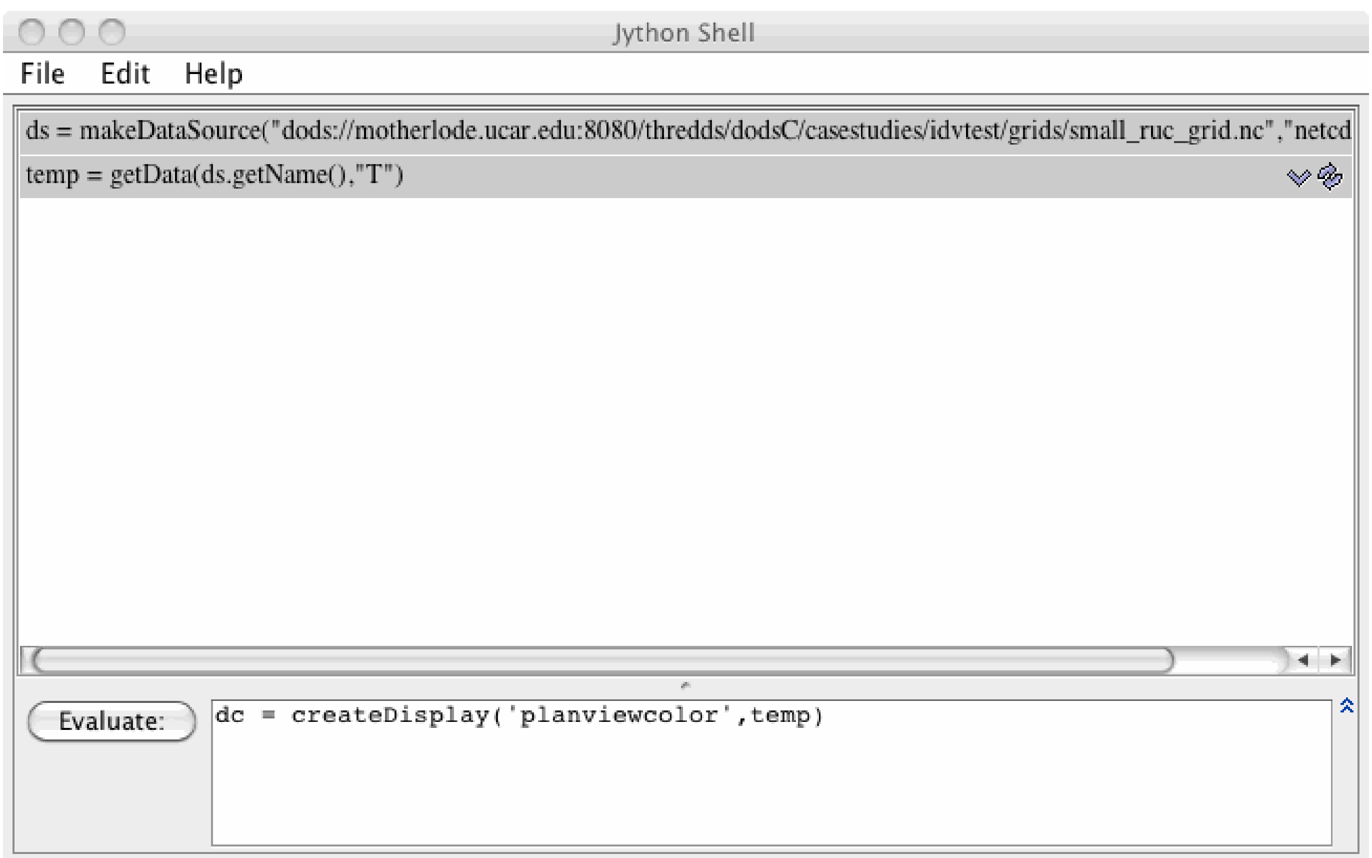
IDV Jython scripting environment to visualize data interactively.
Leveraging the IDV Jython command shell with VisAD creates a powerful combination for scripting IDV workflows as well as "experimental exploration" of data and the IDV and VisAD APIs. In this section, we will examine to how make use of the IDV Jython scripting environment to visualize data interactively.

To achieve this objective, we will leverage the VisAD Jython APIs. Complete documentation for these APIs can be found here:

- JPythonMethods
- JythonLib

The JPython (the old name for Jython) VisAD API is a low-level API for interacting with VisAD objects from the Jython environment. The IDV JythonLib is a higher-level API built on top on the VisAD JPython API. In the following exercises we will be using both these APIs.

These exercises are meant to be done in sequence. We will be using the Jython interactive shell. Please enter the commands described below in the evaluation area as shown in the figure.



1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Select the **Edit Formulas Jython Shell** menu item.
3. Write a script to load some RUC data and display it in the planview. Copy this text into the Jython Shell to the right of the evaluate button and click on the `Evaluate` button.

   ```
   ds = makeDataSource("dods://motherlode.ucar.edu:8080/thredds/dodsC/casestudies/idvtest/gri
   temp = getData(ds.getName(),"T")
   dc = createDisplay('planviewcolor',temp)
   ```

   This script should result in a color shaded plan view of temperature data.
4. Have the IDV prompt the user for the data interactively.

   ```
   rh = selectData()
   ```

   You will be prompted for data. Select relative humidity.

   ```
   dc = createDisplay('planviewcolor',rh)
   ```
5. We will now write a couple of convenience methods

   ```
   def retrieveData(dset, field) :
     return getData(dset.getName(),field)

   def createColorShadedPlanView(field) :
     return createDisplay('planviewcolor',field)
   ```
6. Obtain the U and V wind fields and display wind speed in plan view

   ```
   u = retrieveData(ds,"u")
   v = retrieveData(ds,"v")
   ```

   Now invoke the windpeed formula you wrote in an earlier

   ```
   showLib()
   ws = windSpeed(u,v)
   dc = createColorShadedPlanView(ws)
   ```

   If you did not complete that exercises, here is the wind speed formula.

   ```
   def  windSpeed(u,v):
     usqd= u*u
     vsqd= v*v
     ws = sqrt(usqd + vsqd)
     return ws
   ```
7. This is a slightly more complicated example involving the comparison and difference observational satellite temperature data with NAM model temperature data

   If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.

   ```
   sds = makeDataSource("/Users/chastang/tmp/sat/GOES-East100.area")
   img = selectData()
   gds = makeDataSource("http://motherlode.ucar.edu/thredds/dodsC/model/NCEP/NAM/CONUS_40km/c
   grid = selectData()
   ```

Here you will be prompted for the field. Select Formulas -> Grid -> Make 2D slice, then pick the 300hPa temperature -- one time only!

```
diff = img - grid
createDisplay('planviewcolor',diff)
```

Exporting to Jython Library

As you gain familiarity with the IDV Jython Scripting environment, you will eventually wish to save and export your methods to the Jython library. For the purposes of illustrating this functionality, in this exercise, we will write a simple method and export it to the library.

1. Select the **Edit** **Formulas** **Jython Shell** menu item.

   Note that it is possible to have several shell windows open at once. This is a nice feature as you develop your Jython methods in one shell, then copy / paste the completed method to the fresh shell for export.
2. Let's write a formulate to calculate the dew point from the temperature, and relative humidity.

   ```
   def dew_c(rh, temp_c):
     from java.lang import Math
     l = Math.log(rh / 100.0)
     m = 17.27 * temp_c
     n = temp_c + 237.3
     b = (l + (m / n)) / 17.27
     dew_c = (237.3 * b) / (1 - b)
     return dew_c
   ```

3. Select the **File** **Export Commands** menu item. This action will prompt you for a procedure name, but leave this field blank and click the `OK` button.
4. You are now in your Jython User library where you should see your method. Click the `Save` button. You now have your Jython method available for later use.

The VisAD Jython shell is a full programmatic environment. We have only scratched the surface of what can be achieved. We encourage users to interactively experiment with the IDV in this manner.

4.3.5 Batch Scripting with the IDV Jython API

IDV Jython scripting to visualize data in batch mode
In addition to interactive scripting, Jython scripts can run in batch mode for automatic generation of images without necessarily having to "manually" invoke an IDV client GUI. These scripts can be useful for generating images via a crontab, for example. Note that when developing your Jython IDV batch script, it is useful to first experiment with them in interactive mode. See the previous section about the Jython interactive shell.

1. In your favorite editor, create a file call script.py and add the following code

```
setOffScreen(1)
idv.getStateManager().setViewSize(java.awt.Dimension(800,600))
ds = makeDataSource("dods://motherlode.ucar.edu:8080/thredds/dodsC/casestudies/idvtest/gri
temp = getData(ds.getName(),"T")
dc = createDisplay('planviewcolor',temp)
pause()
image = getImage()
writeImage('/home/idv/isl/image.png')
```

2. Now run that script with the following command

```
runIDV -islfile script.py
```

4.4 Configuring IDV Sites

In this section we will cover how to configure the IDV to share common resources within a site installation.

4.4.0 Plugin Manager

We're going to look at how to install and manage IDV plugins, a much easier way of sharing IDV configurations.

1. Bring up the Plugin Manager with the **Tools Plugin Manager** menu.
2. Note the "Workshop IDV" plugin. Click on its button to view the contents.
3. Let's install some of these plugins
    - Under "Displays" install the "User Interface Skins" plugin.
    - Under "Color Tables" install the GEMPAK color tables.
    - View the contents of the GEMPAK color tables. Find the gempak.rbi in the list and view its contents by clicking on its button.
    - Shut down the IDV and restart.
    - Go to the `New Windows` tab under `Quicklinks`. Note the new windows. These came from the skins plugin.
    - Delete the User Interface Skins plugin.
4. The IDV has some support for multiple languages.
    - In the Plugin Manager, click File > Install Plugin From URL, enter the Spanish plugin url

      ```
      "http://www.unidata.ucar.edu/software/idv/plugins/language/es.pack"
      ```

      or French plugin url

      ```
      "http://www.unidata.ucar.edu/software/idv/plugins/language/fr.pack"
      ```

      or Chinese plugin url

      ```
      "http://www.unidata.ucar.edu/software/idv/plugins/language/zh.pack"
      ```
    - Exit and restart the IDV.
    - Exit and restart the IDV with the `-noplugins` argument.
    - Uninstall the Language Pack plugin
5. We also have a simple IDV.
    - Under "Customized IDVs" install the "Simple IDV"
    - Exit, restart and note the changes.
6. So, again you might ask "But Jeff, how do I take all of the resources I've created in the IDV and turn them into one of these plugins?"
7. I'm glad you asked that question. Next page...

4.4.1 Plugin Creator

We're going to look at how to create IDV plugins.

1. Bring up the Plugin Creator with the **Tools Plugin Creator** menu.
2. Add in all of the resources you've created.
   - Go to the **File Add File** menu.
   - This shows the `/home/idv/.unidata/idv/DefaultIdv` directory.
   - Shift-click or control-click to add the resource files, e.g., `stationmodels.xml`, `colortables.xml`, `derived.xml`, etc.
3. Add some items to the menu bar
   - Bring up the `Toolbar` tab in the user preferences (**Edit Preferences**).
   - Select some items on the right and do `Export to Menu Plugin`. Enter some menu name.
4. Go to the `Excludes` and select a number of system resources to exclude. e.g., Color Tables, Station Models, etc.
5. View the `Properties` tab, scroll down and change `IDV Title` to something else.
6. Enter `myplugin.jar` in the file browser and click on `Write Plugin`.
7. Let's see the results.
   - Shutdown the IDV.
   - Bring up a terminal window and do:

     ```
     runIdv -userpath ~/otheridv -plugin /home/idv/myplugin.jar
     ```

     This brings up the IDV with a different local user resource directory.
   - View some of the resources (e.g., Color Table Editor, Station Model Editor)
   - Exit the IDV
   - From a terminal window view the contents of the plugin:

     ```
     jar -tvf myplugin.jar
     ```
8. Lets create another plugin that is something that might be used in a classroom setting
   - Add a page for the quicklinks. Bring up a text editor and create a descriptive html page. You can include links to IDV bundles in this html. Paste this into the file:

     ```
     <a href="http://www.unidata.ucar.edu/software/idv/data/marsquad.xidv"> Load M
     <p>
     <a href="http://www.unidata.ucar.edu/software/idv/data/davenportradar.xidv">
     <p>
     <a href="action:new.displaywindow">New Window</a>
     <p>
     <a href="action:bundle.save">Save Bundle</a>
     <p>
     <a href="action:exit">Exit the IDV</a>
     ```
   - Save this file as `workshop.qhtml`
   - From the Plugin Creator **File Add File** menu select this qhtml file.
   - Add in some of the favorite bundles you've created.
     - ◊ From the Favorites Manager select multiple favorite bundles (Control-Click).
     - ◊ Choose **File Export to Plugin**
   - Write out the plugin, selecting the `Install` checkbox.
   - Exit and restart the IDV

4.5 Miscellaneous Items

4.5.0 Location Files

The IDV supports a variety of location text file formats. All of these formats can be loaded through the File Chooser. Select the "Location" data type.

4.5.0.0 Locations XML

The main one format the IDV uses is a custom locations xml format. This format still has some nomenclature from when it was atmospheric science related (e.g. "stations"). A simple example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<stationtable name="Example">
    <station name="station 1"  lat="65:02:06" lon="-147:30:06" elev="790"/>
    <station name="station 2"  lat="40.3" lon="-107.5" elev="10"/>
    ...
</stationtable>
```

The *lat* and *lon* attributes can be of the form:

```
+/-  ddd:mm, ddd:mm:, ddd:mm:ss, ddd::ss, ddd.fffff ===>   [+/-] ddd.fffff
+/-  ddd, ddd:, ddd::                               ===>   [+/-] ddd
+/-  :mm, :mm:, :mm:ss, ::ss, .fffff                ===>   [+/-] .fffff
+/-  :, ::                                          ===>       0.0
Any of the above with N,S,E,W appended
```

The *elev* attribute is optional. By default it is in meters. You can override this default with an *elevunit* in the stationtable tag. e.g.:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<stationtable name="Example" elevunit="feet">
  <station name="station 1"  lat="65:02:06" lon="-147:30:06" elev="5340"/>
...
```

The *station* tags can have an *id* attribute as well:

```
<station  id="APD" name="Fairbanks/Pedro Dome"
          lat="65:02:06" lon="-147:30:06" elev="790"/>
```

The *station* tags can also have any other attributes:

```
<station  id="APD" name="Fairbanks/Pedro Dome"
          st="AK" co="US"
          lat="65:02:06" lon="-147:30:06" elev="790"/>
<station id="FTG" name="Denver/Boulder"
         st="CO" co="US"
         lat="39:47:12" lon="-104:32:45" elev="1675"/>
```

These can be displayed by the station model used in the Location Display Control.

If you want to create a permenant list of stations that shows up the the **Display Locations** menu, you can create a file called userstations.xml in the `.unidata/idv/DefaultIdv` directory under your user's home directory. The format would look like:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<stationtables>

<stationtable name="SubsetA" category="My Custom Stations">
(list of stations for subset A)
</stationable>

<stationtable name="SubsetB" category="My Custom Stations">
(list of stations for subset B)
</stationtable>
</stationtables>
```

This will show up in the **Display Locations** menu as a top menu called **My Custom Stations** and sub menus **SubsetA** and **SubsetB**.

4.5.0.1 CSV Format

Location data can be defined in a CSV (Comma Separated Value) format. The first line is a comma separated list of column names. There must be columns that are latitude and longitude. These are denoted with (case insensitive):

```
latitude
lat
longitude
lon
long
```

Altitude is given by the column names:

```
alt
altitude
```

The altitude value, if defined, is by default in meters. You can optionally specify a unit with the suffix: "[unit name]" (see example).

The first column that is found that is not one of the location columns is taken to be the name of the location.

Example:

```
Name,Latitude,Longitude,Altitude,State
Boulder,40,-107,5430[feet],CO
Miami,30,-95,0[feet],FL
...
```

4.5.0.2 GeoRSS Format

The IDV can also read GeoRSS formats. This is a geocoded RSS feed. An example is the USGS earthquake feed:

http://earthquake.usgs.gov/eqcenter/recenteqsww/catalogs/eqs7day-M2.5.xml

4.5.1 Text Point Data

If your point (in situ) data is not available in one of the file formats that the IDV can handle, but you can output it into a column-oriented ASCII text file, you may still be able to use the IDV displays. The IDV supports reading in point observation data in a text comma separated value (CSV) file format.

You need to specify the metadata of this point data. This can be done in two ways. First, the file can have two extra header lines that defines for the IDV the fields that are within the file and the types of the fields. The only other requirement is that there must be a latitude, longitude and time field in the data. See below.

Secondly, if there are no metadata header lines defined, the Text Point Data Source will show the Metadata GUI described below.

Examples:

Comma separated numeric values

In the simplest form, each line line is an observation at one time and one location:

```
(index -> (Time,Latitude,Longitude,Altitude,PSL,GUS,SPD,DIR,TD,T))
Time[fmt="yyyy-MM-dd HH:mm:ss"],Latitude[unit="deg"],Longitude[unit="degrees west"],Altitude[unit
t="deg"],TD[unit="celsius"],T[unit="celsius"]
2007-01-07 16:00:00Z,32.9,117.1,145.0,1026.1,NaN,0.0,0.0,-2.8,12.8
2007-01-07 16:00:00Z,48.3,92.5,341.0,1003.7,NaN,1.5,170.0,-2.0,-0.99
2007-01-07 16:00:00Z,36.8,98.7,449.0,1024.0,12.4,9.8,330.0,-3.0,3.0
2007-01-07 16:00:00Z,44.3,121.2,938.0,1030.1,NaN,2.1,110.0,-3.3,-1.7
```

Note also that the first line's structure is quite rigid -- you must have a variable (e.g., **index**, **recNum**) that is the domain parameter; this should map to the range values. The second line defines the formatting and units of the parameters. Unit names should be standard international unit specifications (udunits compatible). A list of valid names can be found here. (A complete description of this format is contained in the VisAD README.text file.)

Also note that you need to set the name of the variable for time as "Time", as well as the locations as "Latitude", "Longitude" and "Altitude" (if needed).

Text fields

If you have text fields (i.e., non-numeric data) in your observation just do this:

```
(index -> (Time,Latitude,Longitude,Altitude,ST(Text),T))
Time[fmt="yyyy-MM-dd HH:mm:ss z"],Latitude[unit="deg"],Longitude[unit="degrees west"],Altitude[un
2007-02-16 11:00:00 MST,32.9,117.1,145.0,CA  ,20.6
```

Here we have a ST field (State from metars). Its field name is defined as "ST(Text)" and its entry in the second line is the same.

Time in multiple columns

If you have time fields that span multiple columns, you can use the `colspan` keyword:

```
(index -> (IDN, Latitude, Longitude, Time, WDIR, WSPD, GST))
IDN, Latitude[unit="deg"], Longitude[unit="deg"], Time[fmt="yyyy MM dd HH mm" colspan="5"], WDIR[
41001 34.68 -72.66 2007 07 17 20 50 210 4.0 6.0
41004 32.5 -79.09 2007 07 17 20 50 210 6.0 MM
41008 31.4 -80.87 2007 07 17 21 50 170 7.0 8.0
```

Skipping columns

You can use the "skip" parameter if you want to ignore (not read) values in your text file that you don't want to use:

```
(recNum->(Latitude, Longitude, Altitude, type(Text), time, turb_intensity))
Time[fmt=yyyyMMddHH], obtime, skip, type(Text), skip, skip, Latitude, Longitude[scale=-1], Altitu
2004050100 0005    34     C210 1 T  38.82   92.22   7000   7000  0 -9
2004050100 0004    35     PA32 0 T  35.40   98.62   4000   4000  0 -9
2004050100 0008    58      A36 0 T  29.18   81.05   5000   5000  2 -9
```

Fixed values for several observations

If you have values (like time or location) that are fixed for several observations, you may use this construct:

```
(index -> (Longitude,Latitude,Time,ST(Text),SPD,DIR,TD,T))
Longitude[unit="degrees west"],Latitude[unit="deg"],Time[fmt="yyyy-MM-dd HH:mm:ss z"],ST(Text),SP

Longitude=-117.1
Latitude=32.9
ST=MSN
2007-02-20 11:00:00 MST  ,0.0,0.0,8.9,13.3
2007-02-20 12:00:00 MST  ,0.0,0.0,11.9,15.0

Longitude=-89.4
Latitude=43.1
ST=DEN
2007-02-20 11:00:00 MST  ,1.5,160.0,-7.0,-2.0
2007-02-20 12:00:00 MST  ,1.5,160.0,-7.0,-2.0

Longitude=-121.2
Latitude=44.3
ST=ORD
2007-02-20 11:00:00 MST  ,10.8,230.0,-1.1,6.7
```

Loading into the IDV Finally, after you have created your file, you will want to tailor your IDV display in two ways:

1. Create a Layout (station) model for those variables using the IDV's menu "Tools->Layout Model Editor"

Then, when you run the IDV to read your data:

1. Use Data Source Type as "Text Point Data files"
2. Use the "Point Data Plot" for the Display
3. Select your newly created Layout model

Using the Text Point Metadata Gui

If your text data does not have the metadata header lines the Text Point Data Source will show the following dialog which allows you to specify the meta data.

Skipping lines
At the top a number of the initial lines from the text data are shown. The arrow keys allow you to specify the start line. For example, if you had some other header information you can skip over those lines.

Specifying metadata
For each column of text data there is a row shown in the bottom of the dialog. This shows the sampled value and allows you to enter a name, unit, date format, missing value and extra information. There are some names that the IDV treats special: "Latitude", "Longitude", "Altitude", and "Time". You must have at least Latitude, Longitude and Time specified.

The Unit/Date Format field allows you to specify the Unit for data fields and the date format. For text fields choose **Misc Text** as the unit.

The Extra fields must be of the form:

```
name="value"
```

Don't forget the quotes!

Skipping columns
You can skip over certain columns by entering the Name: "skip"

Saving this as a preference
After all of your hard work to keep from having to do this again next time you load a new text point data of the form simply press the "Preferences" button and select **Save Current**. This allows you to save these metadata settings and reapply them later using the "Preferences" button.

4.5.2 ximg File Format

The Image Xml file format (.ximg) allows one to define collections of geolocated images (and also shapefiles). The simplest file can define one geolocated image:

```
<image url="sboulder.jpeg"
     name="South Boulder-aerial photo"
     ullat="39.98890" ullon="-105.22782"
     lrlat="39.98755" lrlon="-105.22548"/>
```

The *url* attribute can be an absolute or relative url or file path. The *ullat*, *ullon*, *lrlat* and *lrlon* attributes are the upper left and lower right lat/lon of the image. It is assumed that the image is in a geographic (i.e., rectilinear, lat/lon) projection.

Collections

You can also define a collection of images:

```
<collection name="Boulder Images">
  <image url="sboulder.jpeg"
       name="South Boulder-aerial photo"
       ullat="39.98890" ullon="-105.22782"
       lrlat="39.98755" lrlon="-105.22548"/>

  <image url="bouldertopo.jpeg"
       ullat="40.06654" ullon="-105.34710"
       lrlat="39.98040" lrlon="-105.19676"
       name="Boulder topo"/>

  <image url="bigtopo.jpeg"
       ullat="40.22807" ullon="-106.66437"
       lrlat="39.54718" lrlon="-105.45623"
       name="Mountains topo"/>
</collection>
```

Collections can contain other collections:

```
<collection name="My Images">
    <collection name="Madison Images">
        <image url="madison_aerial.jpeg"
              ullat="43.09444" ullon="-89.52626"
              lrlat="43.01143" lrlon="-89.36579"
              name="Madison aerial"/>

        <image url="madison_topo.jpeg"
               ullat="43.09444" ullon="-89.52626"
               lrlat="43.01143" lrlon="-89.36579"
               name="Madison topo"/>
    </collection>
    <collection name="Boulder Images">
       <image url="sboulder.jpeg"
              name="South Boulder-aerial photo"
              ullat="39.98890" ullon="-105.22782"
              lrlat="39.98755" lrlon="-105.22548"/>
     </collection>
```

```
</collection>
```

There is also a *shape* tag for defining shape files:

```
<collection name="Shapes">
   <shape url="boulder_roads.zip" name="Boulder roads"/>
   <shape url="boulder_rivers.zip" name="Boulder rivers"/>
</collection>
```

Grouping

The *group* tag allows you to group a set of images in time or space:

```
<group name="group of images"
       format="yyyyMMddhhmm"
       ullat="39.991856"
       ullon="-105.226944"
       lrlat="39.989426"
       lrlon="-105.222656">
       <image
           date="200610011000"
           url="sketch.jpg"/>
       <image
           url="map.jpg"
           date="200610011100"/>
</group>
```

The *format* attribute defines the date/time format of the *date* attributes. The location attributes (ullat,ullon, etc.) can be defined both in the *group* tag as well as in each individual *image* tag.

If there are no *date* attributes then the group of images are aggregated together, each potentially covering a different area.

Images in 3D Space

One is not limited to just specifying the positions of the upper left and lower right corners of the image. There is support for defining the latitude and longitude for any of the four corners of the image:

```
  <image url="image.jpeg"
       name="Image"
       ullat="40" ullon="-100"
       lllat="30" lllon="-100"
       urlat="50" urlon="-90"
       lrlat="30" lrlon="-90"
      />
```

One can also specify the altitude of any of the points. The unit specification is not required and will default to meters.

```
  <image url="image.jpeg"
       name="Image"
       ullat="40" ullon="-100" ulalt="20000\[feet]"
       lllat="30" lllon="-100" llalt="0\[feet]"
```

```
     urlat="50" urlon="-90"  uralt="5000\[feet]"
     lrlat="30" lrlon="-90"  lralt="0\[feet]"
    />
```

The IDV will try to fill in defaults. So, for example, if you wanted to have an image be a vertical cross section you could do:

```
<image url="topo.jpg"
     name="Image"
     ullat="40" ullon="-100"
     urlat="50" urlon="-90"
     ulalt="20000\[feet]"
     llalt="0\[feet]"/>
```

Here, we define the lat/lon of the upper left and upper right of the image. We define the altitude of the upper left and lower left corners of the image. The altitude of the upper right corner defaults to that of the upper left and the altitude of the lower right defaults to the lower left altitude.

4.5.3 Image Movie Format

The IDV can display a sequence of time-stamped images as an animation. This is most commonly used with the Web-Cam Display Control.

One can write their own xml file and display the images within the IDV using this xml format.

There are really two xml formats used. The first defines a set of *imagesets*:

```
<imagesets base="http://www.unidata.ucar.edu/georesources/webcams/images" name="IDV Webcam
    <group name="Rockies">
        <imageset name="Boulder, CO"
                  index="boulder_co/index.xml"
                  lat="40.0" lon="-105.27"/>
        <imageset name="Denver, CO"
                  index="denver_co/index.xml"
                  lat="39.75" lon="-105"/>
    </group>
    <group name="National Parks">
        <imageset name="Theodore Roosevelt National Park, ND"
                  index="theodorerooseveltnationalpark_nd/index.xml"
                  lat="46.94889" lon="-103.43306"/>
        <imageset name="Big Bend National Park,TX"
                  index="bigbendnationalpark_tx/index.xml"
                  lat="29.25" lon="-103.25"/>
        <imageset name="Olympic National Park"
                  index="olympicnationalpark/index.xml"
                  lat="48.26667" lon="-124.675"/>
    </group>
    ...
</imagesets>
```

This is a "table of contents". The *base* attribute, if defined, is used as a url base to prepend to the urls defined by the *index* attributes. This file can be imported by the Web-Cam Display Control. The *lat* and *lon* attributes are optional and are used to locate the source of the movie on a map for the user to select.

Each of the *index* attributes refers to an *images* xml file of the form:

```
<images base="http://www.unidata.ucar.edu/georesources/webcams/images/boulder_co"
        name="Boulder, CO"
        group="Rockies"
        format="yyyyMMddHHmmz" desc="From: http://9news.com">
    <image time="200607251446GMT" file="image_200607251446GMT.jpeg"/>
    <image time="200607251430GMT" file="image_200607251430GMT.jpeg"/>
    <image time="200607251414GMT" file="image_200607251414GMT.jpeg"/>
    <image time="200607251357GMT" file="image_200607251357GMT.jpeg"/>
    <image time="200607251344GMT" file="image_200607251344GMT.jpeg"/>
    ...
</images>
```

The *images* tag defines a *base* attribute (optional, used to prepend to any image urls), a *name*, a (optional) *group* and a date *format* and a description (*desc*). Each *image* tag has a *time* in the format specified in the *images* tag and a *file* attribute which refers to some image. This may be an absolute or relative url or file path.

This file can also be directly imported by the Web-Cam Display Control.

4.5.4 Performance Tuning

If you are running into issues with memory consumption or slow response of the IDV, there are several things you can do.
The amount of memory used by the IDV will depend on the size of the datasets you use and the types of displays. Datasets rendered as 2D depictions (plan views - contours or color shaded displays) use much less memory than 3D displays (isosurfaces, cross sections). Large datasets (images, dense grids) will use much more memory.

There are several features in the IDV that allow you to more efficiently view large datasets:

4.5.4.0 Temporal/Spatial Subset of Data

Some data sources allow you to subset the data temporally and spatially. You can set these properties for all fields in a dataset through the **Properties** menu of the data source (double click on the Data Source in the `Field Selector`) or you can set these for individual field using the tabs in the lower right corner of the `Field Selector`. For more information, see the Data Source Properties section of the IDV User's Guide.

4.5.4.1 Memory allocation

By default, the IDV startup script (**runIDV** (Unix) or **runIDV.bat** (Windows)) tunes the amount of memory allocated to the IDV according to system parameters. On 64 bit computers, the memory allocation amount is 80% of the available RAM. On 32 bit computers, the amount is the minimum of 1.5GB and the available RAM minus 512MB.

In addition, users can change the memory settings in the **Edit** **Preferences**, **System** tab. In order for these changes to take effect, the user will have to restart the IDV.

In rare circumstances, the IDV start script cannot determine the optimal amount of memory for the IDV. In this unusual case, the IDV start script allocates 512MB. The user can still go to the **Edit** **Preferences**, **System** tab and adjust memory settings. Again, the user must restart the IDV in order for this to take effect.

In other exceptional situations, the user may still wish to override the automatic tuning mechanism. In the body of the runIDV script, there are instructions on how to achieve this change, although this should rarely be necessary.

4.5.4.2 Data Caching

By default, the IDV caches the data used for a display in memory. If a field is used more than once for several displays, caching the data prevents an additional reading from of the data from disk or a remote server. If you are only displaying/using a field (i.e. not using it for multiple displays or calculations), you can keep the IDV from caching it in memory. You can turn off data caching by unchecking the `Cache data in memory` checkbox on the `System` tab of the user preferences (accessible from the **Edit Preferences** menu).

4.5.4.3 Caching to disk

The IDV has a caching facility where actively used data (e.g., gridded fields, satellite image, radar) is held in a memory cache. As the amount of data increases the IDV will write the data out to a temporary space on your local disk. If that data is needed again (e.g., rerendering the display) then the IDV will need to go to disk and re-read the data. This may cause some delays.

The memory cache size is intially set at 30% of your maximum memory. This can be changed in the **Edit Preferences**, **System** tab.

In a worst case scenario you could have a very long animation loop of imagery. In this case every time one of the images is displayed while animating its data needs to be accessed. If you have very large images or a very long loop then the images needed to display will be on disk and the time it takes to read them from disk for display will be quite noticable. In this case you can reduce the resolution of the images, reduce the number of times being displayed or increase the cache and/or overall memory size.

4.5.4.4 Maximum grid/image size

You can also set the maximum size of a grid or image that will be displayed. This will allow you to download a large image or grid, but it will be re-sampled before displaying if it is larger than the maximum size you have asked for. You can set the maximum image/grid size under the `System` tab of the user preferences (accessible from the **Edit Preferences** menu).

4.5.4.5 Fast Rendering

By default, the IDV will NOT try to adjust the data renderings to account for projection seams. This is computationally intensive in some cases and slows down the display of data. When the preference "Use Fast Rendering" (under the `General` tab of the user preferences (**Edit Preferences** menu)) is set, the IDV will not try to account for the projection seams. If you are displaying data in its native projection, this will result in faster rendering of the data depiction. However, if you have several displays of data, each from a different data source and on a different projection, you may see anomalies in the displays (spurious lines, portions of images). At that point, you can turn off fast rendering for a particular display using the **Edit Properties** menu of the Display Control for that display, or set your system preference back to not use fast rendering.

4.5.4.6 Parallel Rendering and Data Reading

If you are running the IDV on a multi-core machine you can configure the IDV to render individual time steps in parallel. You can also do remote data reads in parallel. This typically results in a 50% reduction in overall data reading and rendering time.

There are 2 preferences in the **Edit Preferences**, **System** tab. One is the number of threads used for rendering. This defaults to the number of processors on your machine. The second is the number of threads used for data reading. This defaults to 4.

For rendering the IDV will render each time step in parallel. Note: since the rendering processes can allocate temporary memory it is possible to exhaust the available memory if too many threads are running concurrently. While we do not get linear speedup with the number of cores available for rendering (probably due to memory contention issues) we do see 40%-50% performance improvements for complex rendering tasks (e.g., contouring).

The second preference is used when reading individual time steps of data from remote ADDE and OpenDAP servers. This parallelization takes advantage of the multiple cores available on the remote server and somewhat the available bandwidth on the network. We do see a linear speed up in accessing remote data based on the number of cores on the remote server (ADDE or OpenDAP). However, we've seen that if you load the server too much your performance is degraded, probably due to file system issues.

5 Java Developer Topics

This section will cover a variety of topics for Java developers interested in using and extending the IDV.

5.0 Development Environment

5.0.0 Building the IDV from Source

Developers can download the source files for the IDV and build it from scratch.
Before building the IDV you need to first install Java and Java 3D on your system. You need to use the Java SDK, because the Java RunTime Environment (JRE) does not contain the Java compiler (javac). You need to use a version of Java 1.5 (preferably the latest version). You can use either the Java 3D SDK or JRE, version 1.3.x.

You will also need to install **Ant** (http://ant.apache.org/) on your machine. Ant is similar to make - instead of a `Makefile`, Ant uses a `build.xml` file. Be sure to follow the instructions for installing Ant, especially those for setting up environment variables.

Download the IDV source files from the Unidata Web Site (under **Downloads->IDV->Source**):

1. Open the IDV Downloads page in a new window by clicking here.
2. To download the IDV source, you must be registered as a Unidata user. If you have already registered, log in using your user name and password. If you have not registered or you have forgotten your password, follow the instructions on the login sidebar.
3. On your computer, create a folder/directory called "idv".
4. Under "IDV 3.1 Documentation and Source Code", click on the IDV 3.1 Source Code link to start the download. You will be prompted for a location to save the file to. Save it to the idv directory you just created.
5. You will also need the set of ancillary libraries (VisAD, netCDF, etc) that the IDV uses contained in the zip file: IDV 3.1 Jar Files. Save this to the idv directory.
6. After the download is complete, close the extra browser window.

You are now ready to install the IDV source.

1. Open a terminal window.
2. Change to the idv directory (if you are not already there).
3. unJAR the source file.

   jar xvf (downloaddir)/idv_src_3.1.jar

   This will create a "src" directory underneath the install directory (where you unJAR'ed the file).
4. Next, you need to create a `src` directory as a peer of the `ucar` directory under `idv` that holds the ancillary JAR files from the `IDV Binary distribution` that are needed to build the classes. The structure will look like this:

```
idv
   |
   +----src
   |
   +----lib (holds the ancillary jar files).
```
5. Change to the `lib` (if you are not already there).

   cd /home/idv/idv/lib
6. Unzip the file you downloaded into this directory.

unzip ../(downloaddir)/idv_jars_3.1.zip

Now you should be able to build the IDV from scratch.

1. From the download directory execute the ant command to build `idv.jar`.

   ant idvjar

   This removes all Java class files (there will be none the first time), re-builds them and creates the `idv.jar` file, copying it to the lib directory.

2. To run the IDV using your build:

   - Change to the `lib` directory

     cd ../lib
   - then run:

     java -Xmx1024m -jar idv.jar

5.0.1 IDV Source Tree

An overview of the major packages of the IDV

- ucar.unidata.idv -- Holds the main application framework.
    - ucar.unidata.idv.ui -- Holds user interface related classes.
    - ucar.unidata.idv.collab -- Handles multi-user collaboration.
    - ucar.unidata.idv.control -- Holds the set of concrete DisplayControl classes.
        - ucar.unidata.idv.control.chart -- Holds the set of classes to support chart displays.
        - ucar.unidata.idv.control.drawing -- Wrappers for glyphs used in the DrawingControl
    - ucar.unidata.idv.resources -- Holds a variety of resources that instantiate an application.
    - ucar.unidata.idv.chooser -- Contains the data chooser implementations.
- ucar.unidata.util -- Holds a set of utilities.
- ucar.unidata.ui -- Holds various user interface facilities.
    - ucar.unidata.ui.colortable -- Color table editor
    - ucar.unidata.ui.drawing -- Generic drawing editor framework
    - ucar.unidata.ui.symbol -- Our station (layout) model editor
- ucar.unidata.xml -- Contains a variety of classes related to xml.
- ucar.unidata.collab -- Interfaces and base classes for sharing state.
- ucar.visad -- Unidata specific code that deals directly with Visad.
    - ucar.visad.display -- Contains DisplayMaster/Displayable.
- ucar.unidata.data -- Holds classes that know how to access and/or read in data sources (e.g., netCDF files, ADDE servers, etc.)
    - ucar.unidata.data.grid -- Deals with grid oriented data.
    - ucar.unidata.data.gis -- Deals with GIS oriented data.
    - ucar.unidata.data.imagery -- Deals with image oriented data.
    - ucar.unidata.data.point -- Deals with point oriented data.
    - ucar.unidata.data.sounding -- Deals with sounding/trajectory data.
    - ucar.unidata.data.text -- Deals with text oriented data.

5.0.2 IDV Libraries

The *lib* module contains all of the external jar files the IDV uses.

- *auxdata.jar* Holds the auxdata module. Icons, maps and user guide.
- *external.jar* Holds a set of class files from a variety of other packages. We use this so we don't have to change classpaths, etc., everytime we add a new package jar.

abom.jar	Australian Bureau of Meteorology analysis routines
commons-fileupload.jar	Apache Commons file upload
commons-net.jar	Jakarta Commons Net
derby.jar	Derby database
derbytools.jar	Derby tools
extra.jar	Miscellaneous .class files
geotransform.jar	Projection code
ij.jar	ImageJ
Jama-1.0.2.jar	A Java Matrix Package
jai_codec.jar	Java Advanced Imaging
jcalendar.jar	Date/Time Chooser
jcommon.jar	JCommon utilities
jfreechart.jar	Charting package
jh.jar	JavaHelp
jmf.jar	Java Media Framework
jnumeric-0.1a3.jar	Numerical Python for Jython
mysql.jar	MySQL JDBC drivers
numericalMethods.jar	Java Tools for Experimental Mathematics
poi.jar	Apache POI for Excel format
postgres.jar	PostGres JDBC drivers
render.jar	Batik (Java SVG Toolkit)
sunrisesunset.jar	Sunrise/Sunset code from ITC

- *local-visad.jar* This allows us to roll out modifications to the visad code before we get a new visad jar file.
- *visad.jar* The main visad jar.
- *ncIdv.jar* The Java/NetCDF code from Unidata's
- *jython.jar* Jython interpreter

5.0.3 IDV Auxdata

The auxdata module contains:

- *docs/userguide* IDV user guide
- *docs/workshop* IDV training workshop
- *docs/developer* IDV developer documentation
- *ui/icons* User interface icons.
- *maps* Default system maps
- *plugins* IDV plugins
- *bundles* Example bundles
- *examples* More example bundles

5.0.4 Building with Ant

Now you should be able to build the IDV from scratch.

1. Change to the `ucar` directory.

    cd idv
2. Execute the ant command to build `clean idv`.

    ant clean idv

    This removes all Java class files (there will be none the first time), re-builds them.

3. To run the IDV using your build use:

    java -Xmx1024m ucar.unidata.idv.DefaultIdv
4. For the purposes of this workshop, we've created an alias so you can just type:

    idvdev
5. List out the CLASSPATH we have set up for you:

    set | grep CLASSPATH

5.0.5 ExampleIdv

We will be using an example application in /home/idv/idv/ucar/unidata/apps/example

- Go to the example directory:

  cd /home/idv/idv/ucar/unidata/apps/example
- To compile just run ant:

  ant

  or do:

  *javac *.java*
- To run the ExampleIdv:

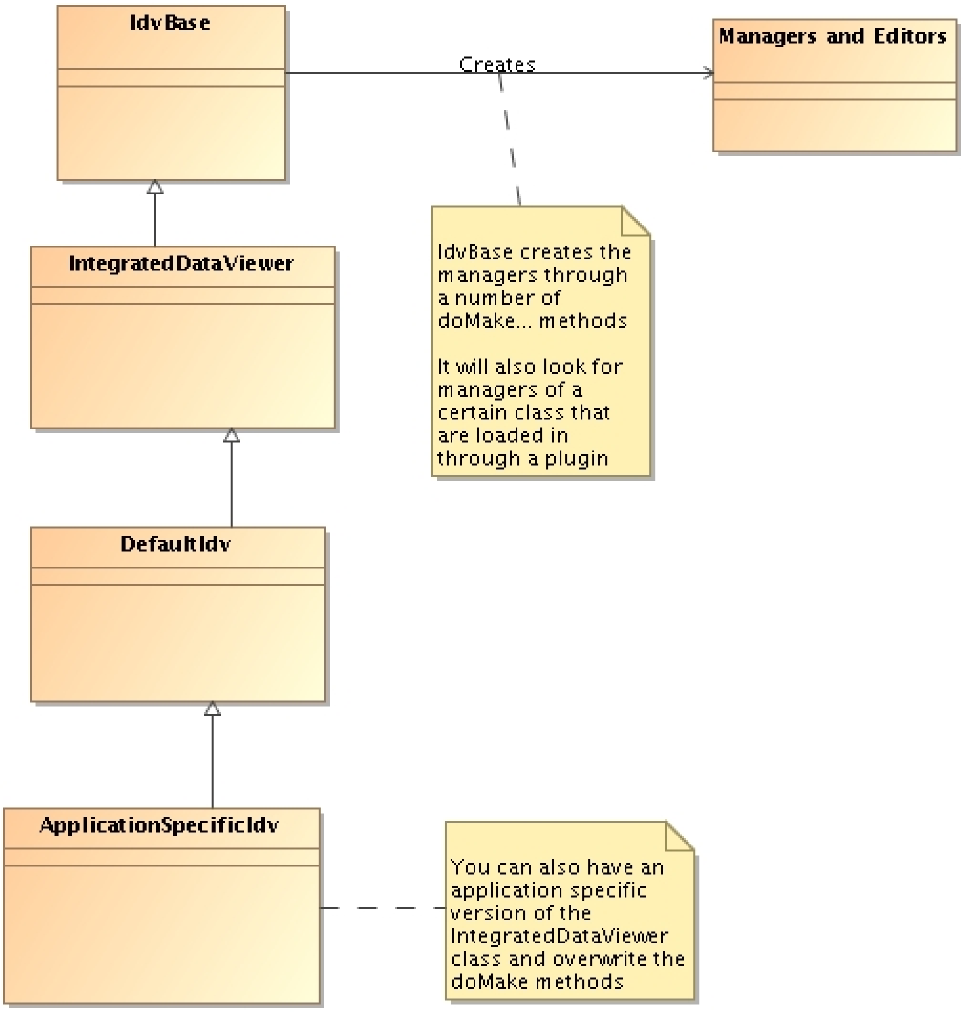
  runExample.sh

  or do:

  java -Xmx512m ucar.unidata.apps.example.ExampleIdv

5.1 IDV Architecture

5.1.0 Core IDV



IdvBase
Creates
Managers and Editors
IntegratedDataViewer
IdvBase creates the managers through a number of doMake... methods
It will also look for managers of a certain class that are loaded in through a plugin
DefaultIdv
ApplicationSpecificIdv
You can also have an application specific version of the IntegratedDataViewer class and overwrite the doMake methods

5.1.1 Managers and Editors

The IDV is a central object for the construction and orchestration of a set of *Managers*, each of which is responsible for different areas of functionality. For example, the ucar.unidata.idv.ArgsManager is responsible for the processing of the command line arguments. The ucar.unidata.idv.IdvResourceManager is responsible for initializing and managing the resources that configure the IDV.

The base class, ucar.unidata.idv.IdvBase contains references to the different managers and methods for accessing and creating them. These methods follow the pattern:

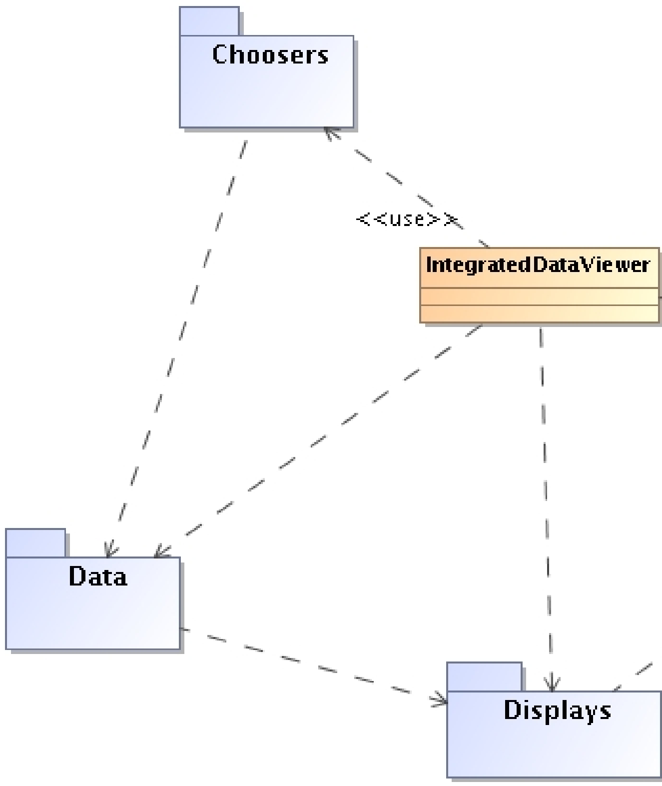
```
DataManager getDataManager()
DataManager doMakeDataManager()
```

ucar.unidata.idv.ArgsManager	Reads in and processes the command line arguments.
ucar.unidata.idv.IdvResourceManager	Handles resources.
ucar.unidata.idv.IdvPreferenceManager	Handles user preferences.
ucar.unidata.idv.IdvPersistenceManager	Bundles.
ucar.unidata.data.DataManager	Data sources.
ucar.unidata.idv.VMManager	View Managers
ucar.unidata.idv.ui.IdvUIManager	User interfaces.
ucar.unidata.idv.StateManager	Properties and other state.
ucar.unidata.idv.JythonManager	Jython.
ucar.unidata.idv.chooser.IdvChooserManager	Choosers.
ucar.unidata.ui.symbol.StationModelManager	Handles the layout models (i.e., station models) defined in the stationmodels.xml and manages the Station Model Editor.
ucar.unidata.idv.DisplayConventions	Provides defaults for how we display data, e.g., what color tables to use, what display unit to use, etc.
ucar.unidata.ui.colortable.ColorTableManager	Color tables. Processes the colortables.xml and manages the Color Table Editor.
ucar.unidata.ui.colortable.ColorTableEditor	The color table editor.

ucar.unidata.idv.ui.AliasEditor	Handles the data aliases (defined in aliases.xml) the Alias Editor.
ucar.unidata.idv.ui.ParamDefaultsEditor	Handles parameter defaults (paramdefaults.xml).
ucar.unidata.idv.ui.AutoDisplayEditor	Not really used right now.
ucar.unidata.idv.publish.PublishManager	Not really used right now.
ucar.unidata.idv.collab.CollabManager	Not really used right now.

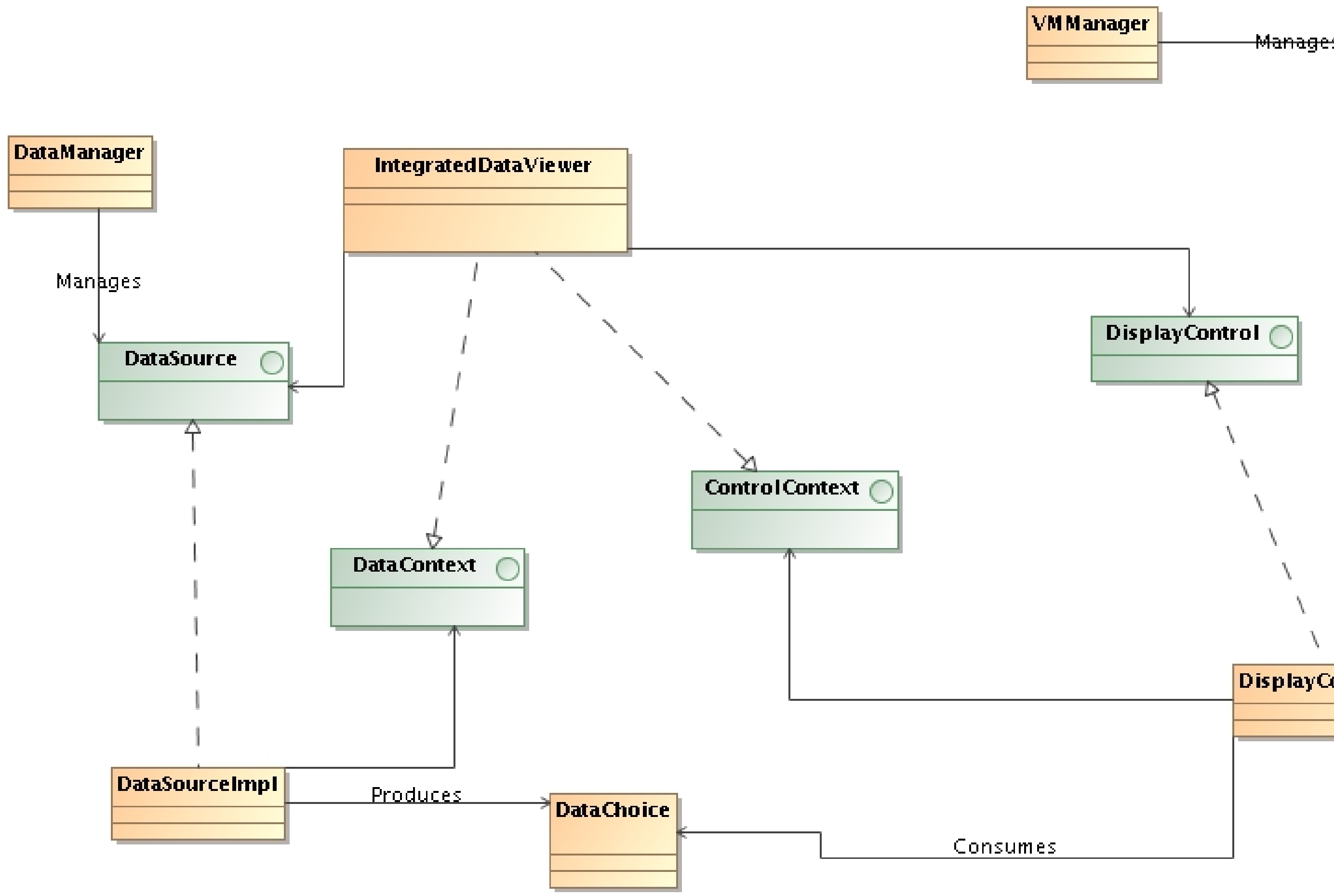
5.1.2 Major Components

The IntegratedDataViewer class serves as the central coordinator for choosing of data, handling of data, displays and views.

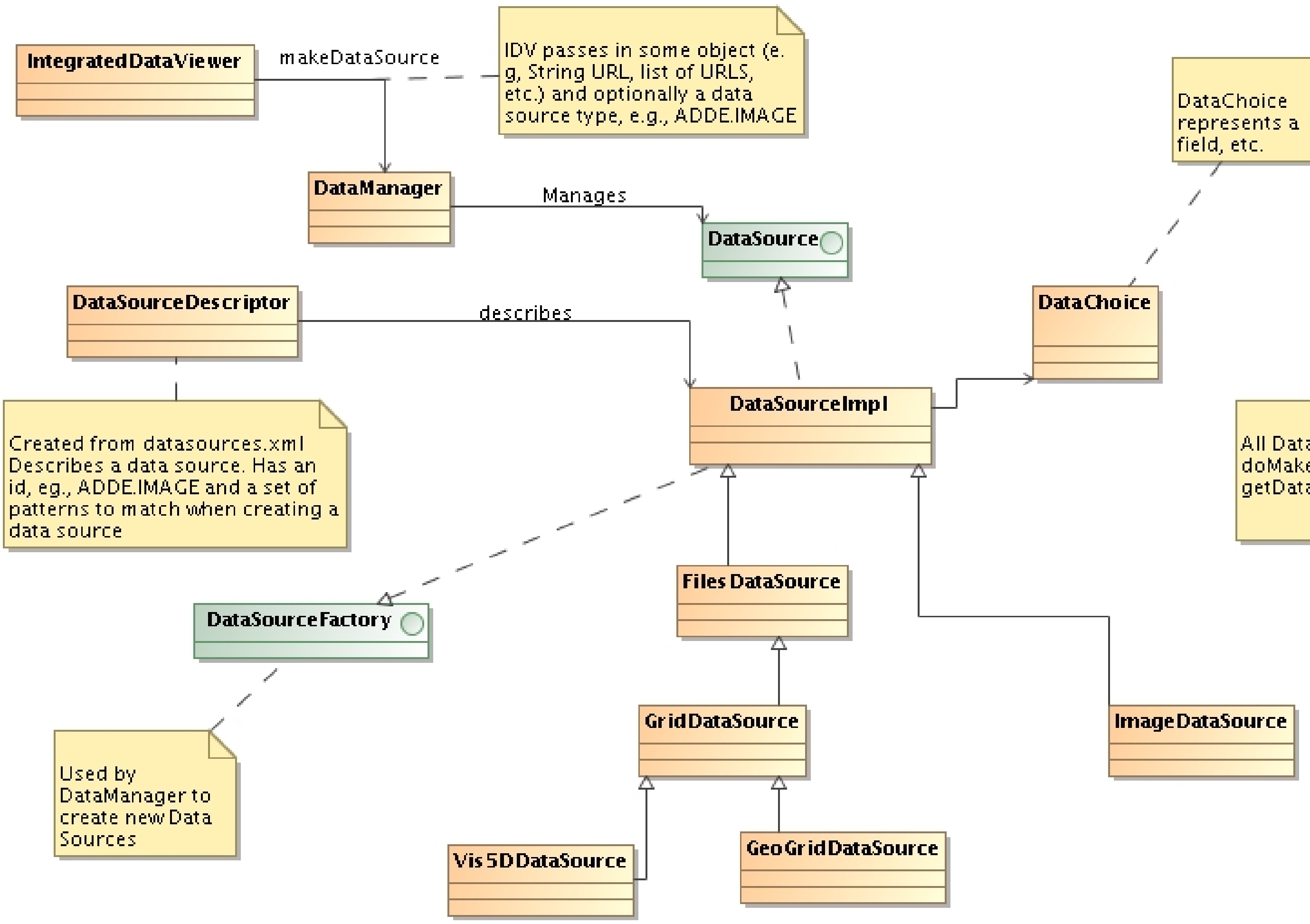


- Choosers select data (e.g., file system, radar chooser).
- The selected files/urls passed to the data sub-system.
- Displays are instantiated with the data.
- Displays are shown in views.

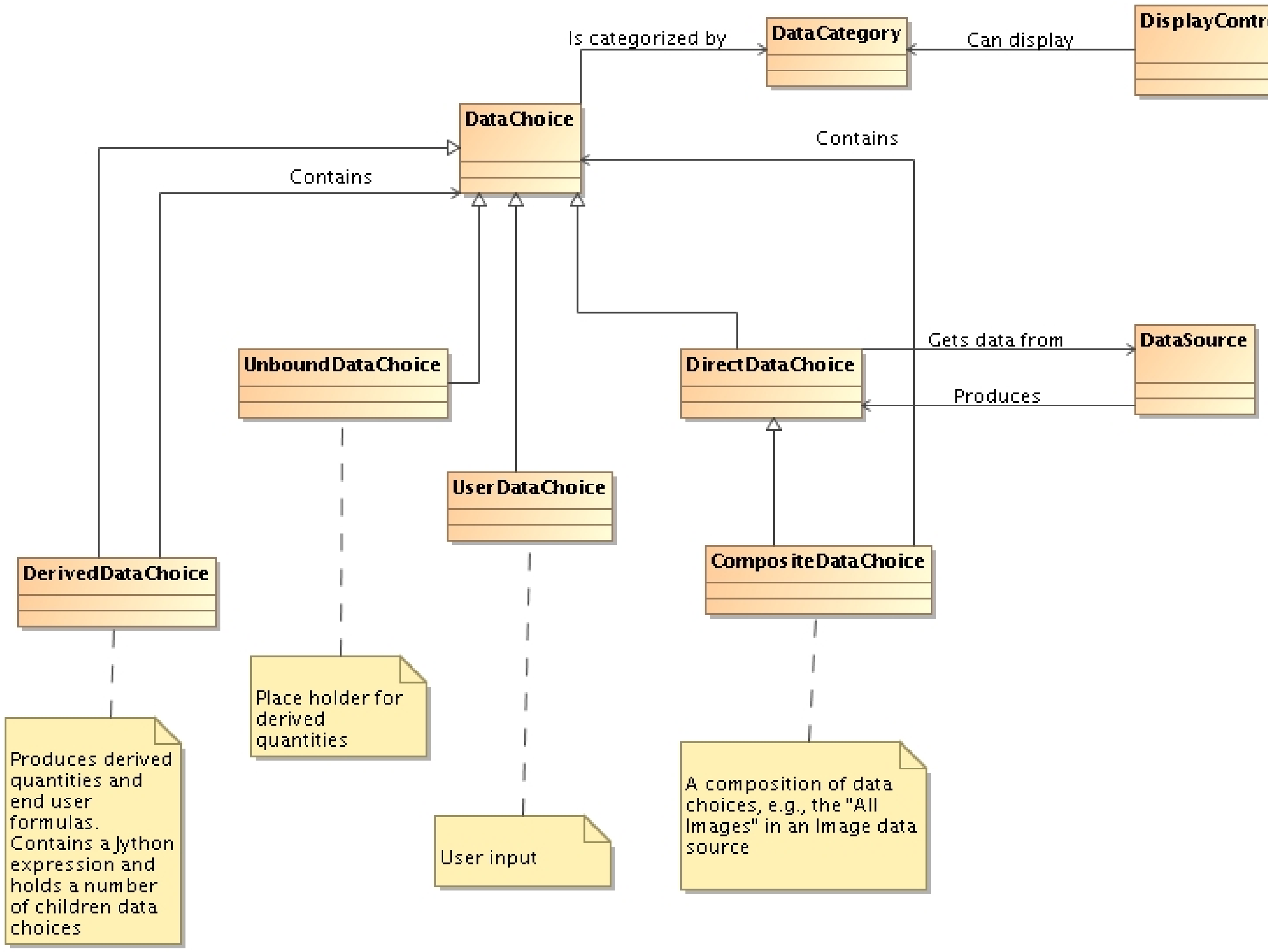
5.1.3 IDV Architectural Overview



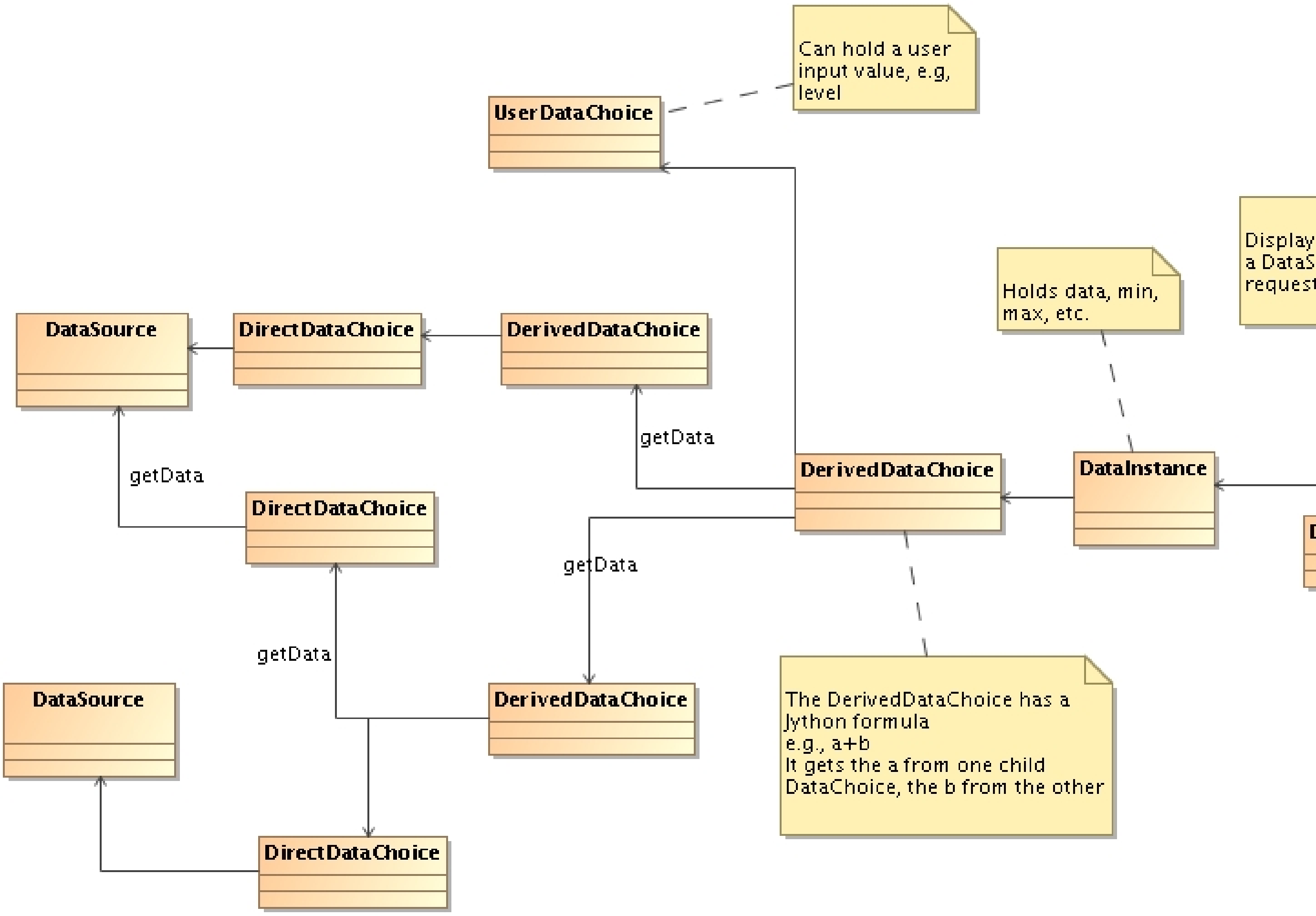
5.1.4 Data Package



5.1.5 Data Choices



5.1.6 Data Flow



5.2 IDV Startup

5.2.0 Initialization and Properties

The IDV starts up with the following sequence:

1. The starting point (e.g., main) of the IDV is an application specific class derived from ucar.unidata.idv.IntegratedDataViewer, e.g., ucar.unidata.apps.example.ExampleIdv.
2. When the IDV starts up a singleton instance of this main class is created and the IntegratedDataViewer.init method is called.

   ```
   public static void main(String\[] args) throws Exception {
       LogUtil.configure();
       ExampleIdv idv = new ExampleIdv(args);
   }
   ```
3. The application is configured by first defining an initial set of property files in the code.

   ```
       //Put the default property file in the list before we parse args
       getArgsManager().propertyFiles.add(
           "/ucar/unidata/idv/resources/idv.properties");
       initPropertyFiles(getArgsManager().propertyFiles);
   ```

   The initPropertyFiles method is called to allow subclasses to modify the list. e.g., from ExampleIdv:

   ```
   public void initPropertyFiles(List files) {
       /*
         files.clear();
         files.add("/ucar/unidata/apps/example/example.properties");
       */
   }
   ```
4. The command line arguments are parsed by the ArgsManager.
5. The class, StateManager, is used to initialize the state.
   - Any property files on the command line are added to the list.
   - All of the properties are merged together.
   - A second pass is done loading any any property files that were defined by the *idv.properties* property.
   - Then any *-Dname=value* command line arguments are added to the properties.
6. Default system properties file.
   - Let's look at the default system properties file: idv.properties.
   - Note: the ".proplabel", ".propdesc", etc., are used in the Plugin Creator.
   - The *idv.properties* property defines the other properties files to use:

     ```
     idv.properties = %APPPATH%/idv.properties;%SITEPATH%/idv.properties;%USERPATH%/idv.p
     ```
   - The macros *%APPPATH%*, *%SITEPATH%*, *%USERPATH%* and *%IDVPATH%* are defined as:

     ```
     ##  %USERPATH%   The file system path of the user's .unidata/idv/application directo
     ##  %SITEPATH%   If defined (usually by the -sitepath argument) the directory path
     ##               or url of where to find site resources
     ##  %IDVPATH%    The internal (to the jars) /ucar/unidata/idv/resources  path
     ##  %APPPATH%    The value of the idv.resourcepath property or the package of the ID
     ```
7. Let's go change the example idv.properties file.
   1. cd to */home/idv/idv/ucar/unidata/apps/example/resources*

2. Copy the file: idv.properties up one level to ucar/unidata/apps/example. This is the APPPATH.
3. Run the Example Idv (runExample.sh)

5.2.1 Command Line Arguments

There are a number of command line arguments for the IDV. To view them provide the argument: -help:

```
idv -help
        -help  (this message)

        -properties  <property file>
        -Dpropertyname=value  (Define the property value)

        -installplugin  <plugin jar file to install>
        -plugin  <plugin jar file, directory, url for this run>
        -noplugins  Don't load plugins

        -cleardefault  (Clear the default bundle)
        -nodefault  (Don't read in the default bundle file)
        -default  <.xidv file>
        -bundle  <bundle file or url>

        -oneinstanceport  <port number> (Check if another version of the IDV is running. If so pa

        -nopref  (Don't read in the user preferences)

        -userpath  <user directory to use>
        -sitepath  <url path to find site resources>

        -nogui  (Don't show the main window gui)

        -data  <data source> (Load the data source)
        -setfiles  <datasource pattern> <semi-colon delimited list of files> (Use the list of fil
        -display  <parameter> <display>
        -islinteractive  <run the isl file in interactive mode
        scriptfile.isl  (Run the IDV script in batch mode)
        -currenttime  <dttm> (Override current time for ISL processing)

        -image  <image file name> (create a jpeg image and then exit)
        -movie  <movie file name> (create a quicktime movie and then exit)
        -imageserver  <port number or .properties file> (run the IDV in image generation server m


        -catalog  <url to a chooser catalog>
        -chooser  (show the data chooser on start up)

        -connect  <collaboration hostname to connect to>
        -server  (Should the IDV run in collaboration server mode)
        -port  <Port number collaboration server should listen on>


        -printjnlp  (Print out any embedded bundles from jnlp files)
        -listresources  <list out the resource types
        -debug  (Turn on debug print)
        -debugmessages  (Turn on language pack debug)
        -recordmessages  <Language pack file to write missing entries to>
        -trace  (Print out trace messages)
        -traceonly  <trace pattern> (Print out trace messages that match the pattern)
```

5.2.1.0 Specifying data source type

If you load in a data source from the command line using the -data argument the idv tries to figure out what type of data it is by looking for patterns in the file or url you specify. If it cannot determine the type the IDV will prompt the user for the type.

You can add in a "type:" prefix to the argument that will specify the type with:

```
idv -data type:somedatatype:the_file_or_url_to_the_data
```

The different values for the "somedatatype" are listed in here.

5.2.2 Example Args Manager

Let's go create our own ExampleArgsManager to parse application specific command line arguments.

1. cd to */home/idv/idv/ucar/unidata/apps/example*
2. Uncomment the the doMakeArgsManager factory method in ExampleIdv.java:

   ```
   protected ArgsManager doMakeArgsManager(String args) {
       return new ExampleArgsManager(this, args);
   }
   ```
3. Look at ExampleArgsManager.java. This handles the argument "-examplearg"

5.2.3 Resources

- The IDV knows very little about everything. All of the things that make up an application are defined by a set of resources. This includes:
    - The user interface skins, menu bar, toolbar.
    - The data choosers, what data sources are available, formulas and derived quantities.
    - What displays are available and the the display resources: color tables, station models, display conventions, etc.
    - ...
- The resources are defined at startup time by the *idv.resourcefiles* property. This points to a list of "rbi" files:

```
idv.resourcefiles=%USERPATH%/idv.rbi;%SITEPATH%/idv.rbi;%APPPATH%/idv.rbi;%IDVPATH%/idv.rb
```

- By default the IDV looks in the USERPATH, SITEPATH, APPPATH and the IDVPATH for the RBI files and loads them in that order.
- Let's go look at the system idv.rbi file in ucar/unidata/idv/resources/idv.rbi
    - The RBI is made up of a number of resources tags, each one defines where the IDV can find the resources files for that particular resource. e.g., by default the xml files that define the color tables are found in:

```
<resources name="idv.resource.colortables">
  <resource location="%USERPATH%/colortables.xml"/>
  <resource location="%SITEPATH%/colortables.xml"/>
  <resource location="%APPPATH%/colortables.xml"/>
  <resource location="%IDVPATH%/colortables.xml"/>
</resources>
```

    - The resource locations can point to different things, not just xml files. It just so happens that most of the resource files are defined using xml.
- In a later section we'll see how to control what resources get loaded when using the "loadmore=false" attribute.
- Let's look at the resources in more detail.
    - Run the ExampleIdv with the argument -listresources to see the list of resources and which ones exist.
    - The X'ed red ones are those that don't exist. The green ones exist.
    - The pattern is used for recognizing resources in a plugin.
    - Note, the paths can be a Java resource, a file or a url. The IDV handles all of these transparently.
    - Let's create a couple of things - a new color table, a new station model and a new formula.
    - Shut down and restart with -listresources again.
- Let's look at the local IDV storage directory.
    - cd to ~/.unidata/idv/DefaultIdv
    - This directory holds everything created by the user in the ExampleIdv. Its name comes from idv.store.name property.
    - Note, there should be colortables.xml, stationmodels.xml and derived.xml files there. These hold the resources you created previously.
    - These can be used in a site path or within a plugin.

5.3 IDV User Interfaces

5.3.0 Xml based UI construction

The ucar.unidata.ui.XmlUi class creates a user interface from a "skin" xml.

Note: You can test the following example skins by:
Right click on the link. Save file to your home directory and run:
java ucar.unidata.ui.XmlUi <skinfile.xml>

Note: The IDV offers a xml utility that formats an xml file in place:
java ucar.unidata.xml.XmlUtil <skinfile.xml>

The xml can take a variety of forms:

```
         <somecomponenttag>
            <some other component tag/>
        </somecomponenttag>
```

e.g.:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel>
  <button label="The Button"/>
</panel>
```

example1.xml

Or more complex with the *skin* outer tag, an inner *ui* and optional components and styles sections:

```
         <skin>
           <ui>
               Some ui xml
           </ui>
           <components>
               components
           </components>
           <styles>
               style definitions
           </styles>
        </skin>
```

The skin xml, either under the ui tag or under the components tag, contains a set of nested container and component tags. The panel tags have a layout attribute:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel
   cols="1"
   layout="gridbag">
  <panel
     bgcolor="red"
     layout="flow">
    <button label="Button 1"/>
    <button label="Button 2"/>
    <button label="Button 3"/>
    <panel
```

```
       bgcolor="green"
       cols="1"
       layout="gridbag">
      <label text="Label 1"/>
      <label text="Label 2"/>
      <label text="Label 3"/>
    </panel>
  </panel>
  <textinput value="text field 1"/>
  <textinput
     rows="2"
     value="text field 2"/>
  <textinput value="text field 3"/>
</panel>
```

example2.xml

Components

Each top-level tag under the components tag has an id attribute. One can then refer to this component in the ui tag with a:

```
<component idref="the component id">
```

This allows one to separate overall layout (defined in the ui) from that of actual components. e.g.:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<skin>
  <ui>
    <panel layout="border">
      <component idref="top" place="North"/>
      <component idref="middle" place="Center"/>
      <component idref="bottom" place="South"/>
      <component idref="left" place="West"/>
      <component idref="right" place="East"/>
    </panel>
  </ui>
  <components>
      <label
         id="top"
         text="Top Label"/>
      <label
         id="middle"
         text="Middle Label"/>
      <label
         id="bottom"
         text="Bottom Label"/>
      <label
         id="left"
         text="Left Label"/>
      <label
         id="right"
         text="Right Label"/>
  </components>
</skin>
```

component.xml

Note: any attributes defined in the component tag in the ui section will overwrite the attributes in the actual tag in the components section.

Supported Tags

```
component
panel
tabbedpane
label
menubar
button
checkbox
textinput
menu
image
```

All tags can have these attributes:

bgcolor, fgcolor - background and foreground color. The value can be a color name, e.g.: red, blue, orange, white, etc. or a single numeric value or a comma separated list of rgb values: e.g.: "250,50,10"

fontsize - specify font size used.

fontface - specify font face used.

Tag: component

The component tag can either have an idref, which points to a component defined in the components section:

```
<component idref="some id in the components section"
           (and any attributes)/>
```

Or it can have an id which should be held within the idToComponent Hashtable which the XmlUi is created with. This allows the use of any application specific Component-s

```
<component id="some id in idToComponent Hasthable"
           (and any attributes)/>
```

Tag: panel

```
<panel layout="border|card|grid|gridbag|inset"
       hspace="int, hor. spacing "
       vspace="int, vert. spacing "
       rows="int"
       cols="int"
       x="int"
       y="int"
       colwidths="int,int,...,int"
       rowheights="int,int,...,int">
```

The panel tags can have any number of children tags. The layout of the children is defined with a "layout" attribute which can be one of: border, grid, gridbag, inset.

layout="border" - java.awt.BorderLayout. The children components of this tag should have a "place" attribute which is one of the java.awt.BorderLayout places: North, South, East, West, Center. e.g.:

```
<panel layout="border" >
   <label id="top" place="North" />
   <label id="bottom" place="South"/>
   <label id="left" place="West"/>
   ...
</panel>
```

layout="grid" This is the java.awt.GridLayout. You can specify a number of rows and/or columns. For example, the following gives 2 rows and 3 columns. The spacing used is defined with: hspace=".." vspace="..." attributes.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel
   cols="3"
   layout="grid"
   rows="2"
   hspace="10"
   vspace="20">
  <label text="label1"/>
  <panel bgcolor="red"/>
  <panel bgcolor="green"/>
  <panel bgcolor="blue"/>
  <label text="a much wider label"/>
  <panel bgcolor="black"/>
</panel>
```

grid.xml This would give single row with multiple columns:

```
<panel layout="grid" rows="1">
```

layout="gridbag" This uses the java.awt.GridBagLayout in a column oriented way. The spacing used is defined with: hspace=".." vspace="..." attributes. You can specify the number of columns in the grid. You can also specify the column and row weights (As a comma separated string of numeric values) that determine stretchiness. e.g.:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel
   bgcolor="white"
   layout="inset"
   space="20">
  <panel
     cols="3"
     colwidths="0,1,0"
     layout="gridbag"
     rowheights="1,0">
    <label text="Row 1 height is stretchy"/>
    <panel bgcolor="red"/>
    <panel bgcolor="green"/>
    <panel bgcolor="blue"/>
    <label text="Middle column width is stretchy"/>
    <panel bgcolor="black"/>
  </panel>
</panel>
```

gridbag.xml

layout="graphpaper" This uses a graph paper layout that allows for exact component placement.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel layout="graphpaper" rows="4" cols="5">
  <label
     text="x:0 y:0 cols:3 rows:1"
     cols="3"
     x="0"
     y="0"
     bgcolor="green"
     align="left"/>
  <label
     text="x:0 y:2 cols:2 rows:1"
     x="0"
     y="1"
     cols="2"
     bgcolor="red"
     align="left"/>
  <label
     text="x:2 y:1 cols:1 rows:1"
     x="2"
     y="1"
     bgcolor="cyan"
     align="left"/>
  <label
     text="x:3 y:0 cols:1 rows:3"
     x="3"
     y="0"
     cols="1"
     rows="3"
     bgcolor="gray"
     align="center"/>
</panel>
```

graphpaper.xml

layout="inset" - This is a simple way to wrap a single child component. The spacing used is defined with: hspace=".." vspace="..." attributes.

Tag: styles

Use the styles section to define classes of components:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<skin>
  <ui>
    <panel
       layout="inset"
       space="10">
      <panel layout="border">
        <panel
           layout="flow"
           place="North">
          <button
             class="button1"
```

```
            label="Button 1"/>
         <button
            class="button1"
            label="Button 2"/>
         <button
            class="button1"
            label="Button 3"/>
       </panel>
       <label
          class="labelstyle"
          id="messagelabel"
          place="Center"
          text="label"/>

<!-- You can also have a parent container with "i:..." attributes.
Children inherit those values  -->
       <panel
          i:fgcolor="blue"
          layout="flow"
          place="South">
         <button label="Button 1"/>
         <button label="Button 2"/>
         <button label="Button 3"/>
       </panel>
     </panel>
   </panel>
  </ui>
  <styles>
    <style
       border="button"
       class="button1"
       fgcolor="red"
       mouse_enter="ui.setBorder(this,etched);"
       mouse_exit="ui.setBorder(this,button);"
       space="2"/>
    <style
       class="labelstyle"
       fgcolor="green"
       fontsize="24"
       space="2"/>
  </styles>
</skin>
```

styles.xml

Tag: tabbedpane

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tabbedpane>
  <label
     fontsize="36"
     text="Tab 1 contents"
     title="Tab 1"/>
  <label
     fontsize="36"
     text="Tab 2 contents"
     title="Tab 2"/>
  <label
     fontsize="36"
```

```
     text="Tab 3 contents"
     title="Tab 3"/>
</tabbedpane>
```

tabbedpane.xml

Tag: menubar

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<menubar label="Commands">
  <menu label="Example Menu">
    <menuitem label="Some menu item"/>
    <menuitem label="Some other menu item"/>
    <separator/>
    <menuitem label="And the last menu item"/>
    <separator/>
    <menu label="Sub menu">
      <menuitem label="Item"/>
    </menu>
  </menu>
  <menu label="Another menu">
    <menuitem
       action="foo"
       label="Another menu"/>
  </menu>
</menubar>
```

menubar.xml

Tag: label

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel
   bgcolor="green"
   layout="border">
  <label
     fgcolor="red"
     fontsize="24"
     fontstyle="italic"
     text="Hello there"/>
</panel>
```

label.xml

Tag: button

```
<button  action="some action"  label="label to use"/>
```

Creates a java.awt.Button. The action (like all actions) can be a semicolon (";") separted list of actions.

Tag: checkbox

```
<checkbox  action="some action"  label="label to use"/>
```

Just like the button tag. However, we convert the itemStateChanged event into an action event and pass it on to the actionListener.

Tag: textinput

```
<textinput rows="optional number of rows"
           cols="optional number of columns"
           value="initial text value"
           action="some action"/>
```

Provides either a TextField or a TextArea depending on the number of rows (the default == 1, which gives a TextField). For TextField-s we add an actionListener if the action attribute is defined.

Tag: menupopup

```
<menu label="Some menu label" image="some image url">
     <menuitem label="some menu item"  action="some action" />
     <menuitem label="some other menu item"  action="some other action" />
     <separator/>
     <menu label="some sub menu">
         <menuitem label="..."  action="..." />
         <menuitem label="..."  action="..." />
     </menu>
</menu>
```

If image attribute is defined creates an image button, else creates a text button. When the button is clicked a menu of menuitems, separators and sub-menus is popped up.

Tag: image

```
<image url="some url"
       width="Optional width of image"
       height="Optional height of image"
       action="If defined the image acts like a button"
       border="Optional true|false">
```

This provides a simple image label or an image button (if action is defined). If it is a button and if border==true then the image is drawn with a border (that changes when clicked).

5.3.1 IDV Skins

We're going to look at how skins work in the IDV now.

- From a terminal window cd to:

  cd /home/idv/idv/ucar/unidata/idv/resources/skins
- View the directory listing.
- View the skin.xml

  java ucar.unidata.ui.XmlUi skin.xml
  less skin.xml

  Note the "idv." tags. We'll look at how these are handled in a bit.
- View the *threeviewskin.xml*

  java ucar.unidata.ui.XmlUi threeviewskin.xml
- List out the skin.xml. Note the attribute in the idv.view tag:
  class=${view_class} attribute.

  When the IDV reads these skin files in it replaces these macros with properties defined in the idv.rbi.
- Look at the "idv.resource.skin" resources entry in the main RBI file:

  less /home/idv/idv/ucar/unidata/idv/resources/idv.rbi
- Now, lets see how the IDV puts this all together
  - Bring up the IdvUIManager.java source code in an editor:

    cd /home/idv/idv/ucar/unidata/idv/ui
    <your favorite editor> IdvUIManager.java
  - Look for the doMakeIdvXmlUi method:

    ```
    protected IdvXmlUi doMakeIdvXmlUi(IdvWindow window, List viewManagers,
                                      Element skinRoot) {
        return new IdvXmlUi(window, viewManagers, getIdv(), skinRoot);
    }
    ```
  - This is a factory method for creating our own XmlUi class. This class handles the special "idv." tags.
  - Look at The IdvXmlUi.java source:

    cd /home/idv/idv/ucar/unidata/idv/ui
    <your favorite editor> IdvXmlUi.java
  - Look at the createComponent method:

    ```
    public Component createComponent(Element node, String id) {...}
    ```

    This is the method that the XmlUI class calls to create the components.

5.3.2 Example UI Manager

We're going to look at how to extend the Example user interfaces.

- From a terminal window cd to the example apps directory:

  cd /home/idv/idv/ucar/unidata/apps/example
- View the idv.rbi file:

  less idv.rbi

  and note the skin entry:

```
<!--The different user interfaces available  -->
<resources name="idv.resource.skin">
  <resource
     location="/ucar/unidata/apps/example/resources/skin.xml"
     label="Example>Example UI">
   <property name="view_class" value="ucar.unidata.idv.MapViewManager"/>
  </resource>
</resources>
```

- We're going to change the example skin.
  - cd into the *resources* directory and bring up the skin.xml in an editor.
  - We're going to add a label above the view. Change the following xml:

    ```
    <idv.view place="Center" class="${view_class}"/>
    ```

    To:

    ```
    <panel layout="border" place="Center">
       <label text="Example Skin" fontsize="24" place="North"/>
       <idv.view place="Center" class="${view_class}"/>
    </panel>
    ```
  - Test this real quickly with the XmlUi:

    java ucar.unidata.ui.XmlUi skin.xml
  - If it is correct then run the example idv:

    java ucar.unidata.apps.example.ExampleIdv
- Note that this skin is not the default.
  - Go up a level to the *example* directory.
  - Edit the *idv.rbi* file and add the property line to the skin entry:

    ```
    <property name="default" value="true"/>
    ```

    e.g.:

    ```
    <resource
       location="/ucar/unidata/apps/example/resources/skin.xml"
       label="Example>Example UI">
     <property name="view_class" value="ucar.unidata.idv.MapViewManager"/>
    ```

```
            <property name="default" value="true"/>
          </resource>
```

- Save the idv.rbi file and run the ExampleIdv again.
- Note that there are still the system skins available.
    - Go back to the example idv.rbi file and add a loadmore="false" to the resources tag:

      ```
      <resources name="idv.resource.skin" loadmore="false">
      ```
    - Run the ExampleIdv again. Note that there is only the example skin.
- Now, one more thing. Lets write a little Java code.
    - Go to the example skin.rbi and change the label tag you added to be *example.label*:

      ```
      <example.label text="Example Skin" place="North"/>
      ```
    - Bring up the ExampleUIManager.java in an editor.
    - Look for the *doMakeIdvXmlUi* method. Notice we create our own IdvXmlUi object. So far it does nothing.
    - Lets add some code that can handle this tag. e.g.:

      ```
      public Component createComponent(Element node, String id) {
          String tagName = node.getTagName();
          if(tagName.equals("example.label")) {
              JLabel label = new JLabel(XmlUtil.getAttribute(node, XmlUi.ATTR_TEXT));
              label.setForeground(Color.red);
              return label;
           }
           return super.createComponent(node, id);
      }
      ```
    - Compile and run the ExampleIdv
- Clean up our changes. Remove the loadmore="false" attribute and the *default* property from the idv.rbi
- One more thing, let's look at the ExampleUIManager again and see how we can skip this whole skin thing if needed.
    - Look at the method:

      ```
          public IdvWindow createNewWindow(List viewManagers, boolean notifyCollab,
                                           String title, String skinPath,
                                           Element skinRoot) {...}
      ```
    - This method is responsible for creating the GUI windows in the IDV.
    - Note the code in there that allows us to skip or not skip the skins:

      ```
              if(!testNewWindow)  {
                  return super.createNewWindow(viewManagers, notifyCollab,title, skinPath,
              }
      ```

5.3.3 Example Menu Bar

We're going to look at how to extend the menu bar.

- In the */home/idv/idv/ucar/unidata/apps/example/resources* directory copy the file defaultmenu.xml up a level to the *example* directory.
- Bring the defaultmenu.xml file into an editor.
- Run the ExampleIdv. Note the new menu entry. For kicks, select **Example** **Show resources**. This shows all of the resources loaded into the ExampleIdv.
- Let's add in a Hello World menu item.
    - Add in the following under the "Example" menu tag.

      ```
      <menuitem
         action="jython:idv.helloWorld();"
         label="Hello World"/>
      ```
    - Run ExampleIdv. Select **Example** **Hello World**
    - We get an error. We need to add the helloWorld method to ExampleIdv:

      ```
      public void helloWorld() {
          System.err.println("Hello World");
      }
      ```

      The method needs to be public.
    - Compile and run again.
- Note that we now have code embedded in our menubar xml file. This might not be good practice. We'll use the "actions"resource to define an action.
    - Copy the file resources/actions.xml to the example directory.
    - This defines an action called "example.helloworld":

      ```
      <action
         id="example.helloworld"
         image="/ucar/unidata/apps/example/resources/HelloWorld.gif"
         description="Call hello world"
         action="jython:idv.helloWorld();"/>
      ```
    - Add the following into the defaultmenu.xml. Note the "action:" prefix.

      ```
      <menuitem
         action="action:example.helloworld"
         label="Hello World from action"/>
      <separator/>
      ```
    - Run ExampleIdv.
    - The example.helloworld action is now also available in the toolbar. Bring up the User Preferences
- So, this added our own menu into the menu bar. How do we not use any of the system menus?
    - Bring up the idv.rbi and change

      ```
      <resources name="idv.resource.menubar">
      </resources>
      ```

      To:

      ```
      <resources name="idv.resource.menubar" loadmore="false">
      ```

```
    <resource
       location="/ucar/unidata/apps/example/defaultmenu.xml"/>
  </resources>
```

The loadmore tells the IDV not to use the system resources.

    - Run the ExampleIdv.
    - Remove the above entries from the example idv.rbi file.

- Now, lets look at the default system menubar in /home/idv/idv/ucar/unidata/idv/resources/defaultmenu.xml
    - Note the "id=" attributes. The IDV uses these ids to know where to dynamically add in different menus, e.g., available windows, display lists, data, etc.

5.4 Data Choosers

5.4.0 Overview of Data Choosers
Data Choosers allow users to select data of a particular type/format an bring them into the IDV. The `ucar.unidata.idv.chooser` package holds the core choosers.

5.4.1 Creating a Weather Text Data Chooser
Let's see how we can access weather text bulletins from an ADDE server.

5.4.0 Overview of Data Choosers

Data Choosers allow users to select data of a particular type/format an bring them into the IDV. The `ucar.unidata.idv.chooser` package holds the core choosers.
Some of the IDV choosers are:

- **FileChooser** - for selecting files
- **PollingFileChooser** - chooser that supports polling files in a directory
- **ImageChooser** - IDV ADDE Image chooser
- **Level2RadarChooser** - NEXRAD Level II files
- **RadarChooser** - ADDE Level III Radar Chooser
- **PointChooser** - ADDE Point obs chooser
    - **RaobPointChooser** - subclass for ADDE Upper Air point data
- **ProfilerChooser** - ADDE Profiler chooser
- **RaobChooser** - Upper Air Sounding Chooser
- **XmlChooser** - chooser for XML documents like THREDDS catalog and WMS servers
- **UrlChooser** - chooser for data residing at URL's

Choosers generally extend from the IdvChooser class and need to implement the following methods:

- **doMakeContents()** - makes the UI
- **doLoadInThread()** - handles the adding of the data

Once a chooser class is created, it can be added to the IDV by creating a choosers.xml file. Let's look at the sample here.

5.4.1 Creating a Weather Text Data Chooser

Let's see how we can access weather text bulletins from an ADDE server.

1. Start the Example IDV
2. From the Dashboard, select the `Data Chooser` tab.
3. Select the `URLs` Chooser. The IDV can use the ADDE URL syntax to access weather text on a remote server.
4. Paste the following URL into the text box:

   ```
   adde://adde.ucar.edu/wxtext?apro=zfp&astn=mkx
   ```

   and click the `Add Source` button.
5. In this case, the IDV handles the URL as an input stream and by default displays it in the TextDisplayControl. However, having to type in an obscure URL syntax is not the ultimate User Interface.
6. Exit the IDV.

Let's create a chooser that allows the user to select from some predefined products.

1. cd to */home/idv/idv/ucar/unidata/apps/example*
2. Open `WxTextChooser.java` in an editor and lets look what's there.
3. Exit the editor and compile the class
4. Copy chooser.xml from the *resources* directory to the current directory.

   cp resources/chooser.xml .
5. Open the file in an editor and look at what's defined.
6. Exit the editor.
7. Start the IDV.
8. From the Dashboard, select the `Data Chooser` tab. Notice that our chooser is there. Select a server, product and station and click the `Add Source` button. Try this with different products. If you know some codes and station ids, try them as well.
9. Now let's look at having a custom Data Source to handle our data.

5.5 Data Sources

5.5.0 Overview of Data Sources

The `ucar.unidata.data.DataSource` interface defines the api through which all data sources are accessed. There is a base class `ucar.unidata.data.DataSourceImpl` that provides a set of base services. There are a variety of DataSource implementations:

- ucar.unidata.data.DataSourceImpl
    - ucar.unidata.data.FilesDataSource
        - ucar.unidata.data.grid.GridDataSource
            - ucar.unidata.data.grid.GeoGridDataSource
                - ucar.unidata.data.grid.DodsGeoGridDataSource
            - ucar.unidata.data.grid.Vis5DDataSource
        - ucar.unidata.data.gis.DemDataSource.java, ucar.unidata.data.gis.GeodasDataSource.java, ucar.unidata.data.gis.KmlDataSource.java, ucar.unidata.data.gis.ShapefileDataSource.java, ucar.unidata.data.gis.WmsDataSource.java
        - ucar.unidata.data.imagery.ImageDataSource
            - ucar.unidata.data.imagery.AddeImageDataSource, ucar.unidata.data.imagery.McIDASImageDataSource
        - ucar.unidata.data.point.PointDataSource
            - ucar.unidata.data.point.AddePointDataSource, ucar.unidata.data.point.NetcdfPointDataSource, ucar.unidata.data.point.TextPointDataSource
        - ucar.unidata.data.radar.RadarDataSource
            - ucar.unidata.data.radar.CDMRadarDataSource
        - ucar.unidata.data.sounding.RaobDataSource,
        - ucar.unidata.data.sounding.TrackDataSource,
            - ucar.unidata.data.sounding.SondeDataSource
        - ucar.unidata.data.storm.StormDataSource
            - ucar.unidata.data.storm.AtcfDataSource
            - ucar.unidata.data.storm.STIDataSource
        - ucar.unidata.data.text.TextDataSource, ucar.unidata.data.text.AddeTextDataSource, ucar.unidata.data.text.FrontDataSource
        - ucar.unidata.data.text.TextProductDataSource,
            - ucar.unidata.data.text.NwxTextProductDataSource
            - ucar.unidata.data.text.AddeTextProductDataSource

Creating Data Sources

The IDV uses the `ucar.unidata.data.DataManager` class to create and manage the set of DataSource-s. This singleton object can be accessed by the `IntegratedDataView.getDataManager` method.

The DataManager knows nothing about the specific DataSource classes. Rather, the set of available DataSource-s is defined within the datasource.xml resource file.

This file has the form:

```
<datasource
       id="EXAMPLE.DATA"
       factory="ucar.unidata.apps.example.ExampleDataSource"
       patterns="\.txt$,\.html$"
       label="Example data source"/>
   ...
```

The DataManager parses the XML and creates a set of `ucar.unidata.data.DataSourceDescriptor` objects which hold this information.

The *patterns* attribute is a set of comma separated regular expression patterns used to determine if this data source is applicable for a given file or url. For example, from the main IDV `ucar/unidata/idv/resources/datasource.xml` file we have the entry for the GeoGridDataSource:

```
<datasource id="FILE.NETCDF"
               factory="ucar.unidata.data.grid.GeoGridDataSource"
               patterns=".nc$,.cdf$"
               label="Netcdf files">
```

This data source type is applicable to anything that ends with .nc of .cdf (The $ used in the patterns represents the end of line character in a regular expression.). The patterns are also used to instantiate a set of PatternFileFilters used when choosing files from the file system.

The factory attribute in the XML is the name of the class that implements ucar.unidata.data.DataSourceFactory. This is the class that is instantiated for a particular data source. The DataManager then asks this class to `getDataSource`. We do this so we could separate out the class that creates a DataSource form the actual DataSource. In practice however, the DataSourceImpl class implements this interface by simply returning itself. i.e., the factory that is created is the actual DataSource.

The important DataManager methods include:

```
DataSource createDataSource (String dataName)
```

Here dataName is typically either a url or a file. The DataManager find the data source entry that which contains a pattern that matches the given dataName. It then turns around and calls createDataSource, as described below, passing in the dataName, dataType (i.e., the id from the XML), and an empty properties table.

```
DataSource createDataSource (Object dataName, String dataType, Hashtable
properties)
```

This method can get called directly by data choosers and the IDV. Here dataName can be anything (a url, a filename, a complex data structure). The argument dataType is an *id* within the datasource.xml file. The properties table allows one to pass in extra information when creating a DataSource.

The DataManager first sees if a DataSource with the given dataName object already exists in its list of created DataSource-s. If it does exists then that previously created DataSource is returned. If not, the DataManager looks up the DataSourceFactory class to instantiate from datasource.xml. The DataManager then tries to find and call a constructor on this factory class with the following signature:

```
(DataSourceDescriptor.class, dataName.getClass(), Hashtable.class)
```

ExampleDataSource.java

So, you're probably wondering how you can create your own DataSource. There is an example implementation in this package: . This class derives from DataSourceImpl and needs to implement just 3 methods:

1. **the constructor**
2. **doMakeDataChoices**
3. **getDataInner.**

This class also has to be registered with the IDV so that when some file or other data source on a network has been selected the IDV knows what DataSource to create. You do this by adding an entry into a datasource.xml:

```
<datasource id="EXAMPLE.DATA"
            factory="ucar.unidata.apps.example.ExampleDataSource"
            patterns="\.txt$,\.html$"
            label="Example data"/>
```

This could be the core IDV resource file (located in `ucar/unidata/idv/resources/datasource.xml`) or in a different file that is included as a resource in your application (like the example `ucar/unidata/apps/example/datasource.xml`).

Creating DataChoice-s

The DirectDataChoice's ctor signature looks like:

```
DirectDataChoice (DataSource dataSource,
                  Object id,
                  String name,
                  String description,
                  List categories,
                  DataSelection dataSelection);
```

Where:
dataSource is this object.
id can be anything, a field name, a data structure that identifies some data, etc.
The name is the short name and the description is the long description used.
categories is a list of ucar.unidata.data.DataCategory objects that represent the categories of data that this DataChoice represents.
The dataSelection (which can be null) allows you to define a set of times for this DataChoice.

So you can do:

```
Object someIdForField1 = "field1";
List categories = DataCategory.parseCategories ("2D grid;GRID-2D-TIME;");
DataChoice dc =
```

```
  new DirectDataChoice (this,
                        someIdForField1,
                        "Field 1",
                        "Some desc", categories,
                         null);
addDataChoice (dc);
```

5.5.1 Using VisAD Data in the IDV

The IDV uses VisAD `visad.Data` objects to model the different types of data.
The VisAD data model was designed to support virtually any numerical data. Rather than providing a variety of specific data structures like images, grids and tables, the VisAD data model defines a set of classes that can be used to build any hierarchical numerical data structures. All these objects are defined in subclasses of `visad.Data` and can be compared and manipulated without converting from one form to another. This page is not meant to be a comprehensive overview of the VisAD data model, but rather to show what `Data` objects are used in the IDV. However, let's look at a good description here.

For more detailed information on the VisAD Data model, see:

- Ugo Taddei's **VisAD Tutorial**.
- The VisAD Data Model Tutorial
- The VisAD Developer's Guide (Data section)

Getting data into the IDV

The **getData()** method of the `DataSource` class returns a VisAD `Data` object. This generalization allows us to easily create new DataSources without having to cast the return to a specific form. The basic task of a developer creating a new DataSource is to transform the raw data into one of the known VisAD Data forms that the IDV uses.

Grids

The IDV models grids as `visad.FieldImpl` objects. A grid can take the form of:

```
   (x,y) -> (parm)
   (x,y) -> (parm1, ..., parmN)
   (x,y,z) -> (parm)
   (x,y,z) -> (parm1, ..., parmN)
   (t -> (x,y) -> (parm))
   (t -> (x,y) -> (parm1, ..., parmN))
   (t -> (x,y,z) -> (parm))
   (t -> (x,y,z) -> (parm1, ..., parmN))
   (t -> (index -> (x,y) -> (parm)))
   (t -> (index -> (x,y) -> (parm1, ..., parmN)))
   (t -> (index -> (x,y,z) -> (parm)))
   (t -> (index -> (x,y,z) -> (parm1, ..., parmN)))
```

In general, `t` is a time variable, but it might also be just an index. In the last 4 examples, index can be an index to a set of radar rays for an RHI, or an ensemble index. The `ucar.unidata.data.grid.GridUtil` and `ucar.unidata.data.grid.GridMath` classes provide utilities for slicing, dicing, querying and performing mathematical operations on data that fits into these forms.

Images (satellite, radar)

Just like grids, images are modelled as FieldImpl-s and generally have the form of:

```
   (x,y) -> (parm)
```

```
  (x,y) -> (parm1, ..., parmN)
  (t -> (x,y) -> (parm))
  (t -> (x,y) -> (parm1, ..., parmN))
```

RGB Images (GIF, JPEG, PNG)

These are also modelled as FieldImpl-s, but have 3 or 4 components:

```
  (x,y) -> (red, green, blue)
  (x,y) -> (red, green, blue, alpha)
  (t -> (x,y) -> (red, green, blue))
  (t -> (x,y) -> (red, green, blue, alpha))
```

Point Data

Point data (METAR, earthquake, lightning) are modelled using the `ucar.unidata.data.point.PointOb` structure which has the methods:

```
    /**
     * Get the location (lat/lon/alt) of the observation.
     * @return georeference location
     */
    public EarthLocation getEarthLocation();

    /**
     * Get the time of the observation.
     * @return  time the observation was taken
     */
    public DateTime getDateTime();

    /**
     * Get the data associated with this observation.
     * @return observed data for this location and time.
     */
    public Data getData();
```

In the IDV, we use the `ucar.unidata.data.point.PointObTuple` to implement this interface as a `visad.Tuple`.

Text

Text data is modelled using the `visad.Text` object which is basically a wrapper for a String.

Map Lines

Map lines are a set of lat/lon and possibly alt points defined by a VisAD `visad.Gridded*DSet` where * is usually 2, but can be 3 if altitude is present. It has a 1D manifold in either case. (Latitude, Longitude) (Latitude, Longitude, Altitude)

Miscellaneous

Most other data in the IDV is modelled using some form of FieldImpl structure. For example, aircraft tracks have the form:

```
   (t -> (x,y,z) -> (parm))
```

but wouldn't be thought of as a grid or image. However, using such a structure allows us to easily compare the values along an aircraft track with colocated points in a grid.

Geolocation

Some data (e.g., Point Data) include the necessary information to geolocate data (Latitude, Longitude, Altitude). Other data (grids, images) may have their spatial extents defined in native coordinates (line/element, row/column). The IDV uses the `visad.CoordinateSystem` class to provide the necessary transforms for geolocation. These provide on-the-fly coordinate transforms.

Time

The IDV uses the `visad.DateTime` class for time animation. DateTime describes time values in seconds since some point. It provides methods for formatting the values to timestamps and creating sets from arrays of DateTime objects.

5.5.2 Creating a Weather Text Data Source

Let's create a custom Data Source for our weather text.

1. Change to the example directory.

   cd /home/idv/idv/ucar/unidata/apps/example
2. Open WxTextDataSource.java in an editor and lets look what's there.
3. Copy `datasource.xml` up from the resources subdirectory and take a look at that.
4. Now, edit WxTextChooser and change:

   ```
           makeDataSource(urls, "FILE.ADDETEXT", ht);
   ```

   to:

   ```
           makeDataSource(urls, "ADDE.WXTEXT", ht);
   ```

   to make the appropriate data source.
5. Compile the chooser and data source.
6. Start the IDV.
7. Select a product from the `Weather Text` chooser and click the `Add Source` button.
8. Notice in the `Field Selector`, that our data choices defined in the data source are listed in the `Fields` panel.
9. Expand the first category and click on `Weather Text Bulletin`. Notice which displays are availble for this parameter.
10. Display the data in the `Text/Html Display`.

Now let's remove the extraneous data choices and try this again.

1. Exit the IDV.
2. Open WxTextDataSource.java in an editor.
3. Remove the section in `doMakeDataChoices()` which adds the extra data choices.
4. Start the IDV and load in some weather text.
5. Notice the change in the `Fields` panel.
6. Save this as your default bundle and then reload it.

5.6 Display Controls

5.6.0 Overview of Display Controls

The `ucar.unidata.idv.DisplayControl` interface defines the API through which all data are displayed. There is a base class `ucar.unidata.idv.control.DisplayControlImpl` that provides a set of base services. A DisplayControl generally has a data depiction and a control window for modifying or interacting with the display. There are a variety of DisplayControl implementations:

- ucar.unidata.idv.control.DisplayControlImpl
    - ucar.unidata.idv.control.GridDisplayControl
        - ucar.unidata.idv.control.PlanViewControl
            - ucar.unidata.idv.control.ColorPlanViewControl
            - ucar.unidata.idv.control.ContourPlanViewControl
            - ucar.unidata.idv.control.FlowPlanViewControl
            - ucar.unidata.idv.control.ImagePlanViewControl
            - ucar.unidata.idv.control.RadarSweepControl
            - ucar.unidata.idv.control.ValuePlanViewControl
        - ucar.unidata.idv.control.CrossSectionControl
            - ucar.unidata.idv.control.ColorCrossSectionControl
            - ucar.unidata.idv.control.ContourCrossSectionControl
            - ucar.unidata.idv.control.FlowCrossSectionControl
        - ucar.unidata.idv.control.ThreeDSurfaceContro
        - ucar.unidata.idv.control.TopographyControl
        - ucar.unidata.idv.control.ProbeControl
        - ucar.unidata.idv.control.VerticalProfileControl
    - ucar.unidata.idv.control.ObsDisplayControl
        - ucar.unidata.idv.control.ObsListControl
        - ucar.unidata.idv.control.StationLocationControl
        - ucar.unidata.idv.control.StationModelControl
    - ucar.unidata.idv.control.BaseImageControl
        - ucar.unidata.idv.control.ImageSequenceControl, ucar.unidata.idv.control.ImageControl
    - ucar.unidata.idv.control.AerologicalDisplayControl
        - ucar.unidata.idv.control.GriddedSoundingControl
        - ucar.unidata.idv.control.RaobSoundingControl
        - ucar.unidata.idv.control.TrackSoundingControl
    - ucar.unidata.idv.control.ProfilerControl
        - ucar.unidata.idv.control.ProfilerMultiStationControl
        - ucar.unidata.idv.control.ProfilerTimeHeightControl
    - ucar.unidata.idv.control.LocationIndicatorControl
    - ucar.unidata.idv.control.MapDisplayControl
    - ucar.unidata.idv.control.NoteControl
    - ucar.unidata.idv.control.OmniControl
    - ucar.unidata.idv.control.RangeAndBearingControl
    - ucar.unidata.idv.control.RadarGridControl
    - ucar.unidata.idv.control.ShapefileControl

Creating Display Controls

The IDV uses the `ucar.unidata.idv.ControlDescriptor` class to create set of DisplayControls-s. The set of available DisplayControls-s is defined within the controls.xml resource file.

This file has the form:

```
<control
   id="wxtextdisplay"
   categories="wxtext"
   class="ucar.unidata.apps.example.WxTextDisplayControl"
   description="Weather Text Display"
   label="Weather Text Display"
   displaycategory="General"
   properties="windowVisible=true; displayName=Weather Text Display;"/>
```

The ControlDescriptor class parses the XML and creates a set of `ucar.unidata.idv.ControlDescriptor` objects which hold this information.

- The *id* attribute is a name that can be used to identify this control for command line and scripting arguments.
- The *categories* attribute lists the DataCategory-s that this control is valid for.
- The *class* attribute gives the class name of the DisplayControl
- The *description* attribute is used for descriptions.
- The *label* attribute is used for labels.
- The *displaycategory* attribute is used for categorizing in legends.
- The *properties* attribute is used for passing in String and boolean properties to the control.

So, you're probably wondering how you can create your own DisplayControl. You need to make a subclass of DisplayControlImpl and override the following methods:

1. **the no arg constructor** - for persistence
2. **doMakeContents** - make the GUI
3. **init(DataChoice)** - sets up the initialization.

5.6.1 Using VisAD Displays in the IDV

The IDV uses the VisAD Display system to render data in displays.

Overview of VisAD Display System

An overview of the VisAD Display system is found here.

DisplayMaster/Displayable

The IDV encapulates the building of displays and data depictions using a set of classes in `ucar.visad.display`.

- The `DisplayMaster` class is a wrapper around a `visad.DisplayImpl` and manages a list of `Displayable` objects. The `ucar.unidata.view.geoloc.MapProjectionDisplay` is the most commonly used DisplayMaster in the IDV.
- The `Displayable` class provides support for encapsulating one or more displayed data objects together with their display-dependent state.
- The `DisplayableData` class holds a `visad.DataReference`, a set of `visad.ScalarMap`-s and a `visad.DataRenderer` that control the depiction of the Data in the display.

  Displayable subclasses used in the IDV include:

  - ucar.visad.display.Displayable
    - ucar.visad.display.CompositeDisplayable
    - ucar.visad.display.DisplayableData
      - ucar.visad.display.RGBDisplayable
        - ucar.visad.display.Grid2DDisplayable
        - ucar.visad.display.IsoSurface
        - ucar.visad.display.TrackDisplayable
        - ucar.visad.display.VolumeDisplayable
        - ucar.visad.display.FlowDisplayable
      - ucar.visad.display.LineDrawing
        - ucar.visad.display.ContourLines
        - ucar.visad.display.MapLines
        - ucar.visad.display.LatLonLines
        - ucar.visad.display.IndicatorPoint
        - ucar.visad.display.TextDisplayable
      - ucar.visad.display.ImageRGBDisplayable
      - ucar.visad.display.ColorScale
      - ucar.visad.display.StationModelDisplayable

ViewManagers and DisplayControls

A ViewManager is a wrapper around a DisplayMaster. A DisplayControl can hold one or more Displayable objects and these can be added to one or more ViewManagers.

MapViewManager Example

5.6.2 Creating a Weather Text Display Control

Let's create a custom DisplayControl for our weather text that will use the category "wxtext" that we created in the Data Source.

1. Change to the example directory.

   cd /home/idv/idv/ucar/unidata/apps/example
2. Open WxTextDisplayControl.java in an editor and lets look what's there.
3. Copy `controls.xml` up from the resources subdirectory and take a look at that.
4. Compile the display control.
5. Start the IDV.
6. Select a product from the `Weather Text` chooser and click the `Add Source` button.
7. Display the data in the `Weather Text` display.

Now let's add in a widget that will highlight some text.

1. Exit the IDV.
2. Open WxTextDisplayControl.java in an editor.
3. Uncomment:
   - the section in `doMakeContents` for the JTextField that allows user input.
   - the line in the default constructor to use the `FLAG_COLOR` attribute.
   - the `getColorWidgetLabel()` method
4. Start the IDV and load in some weather text.
5. Type in some text to hightlight and hit enter. Does it get highlighted?
6. Notice that we have a new item in the Edit menu which allows us to change the highlight color. Change it. Does anything happen? Hit return in the TextField. Does it change then?
7. Exit the IDV.
8. Open WxTextDisplayControl.java in an editor.
9. Uncomment the applyColor method. This is called when the color changes.
10. Compile and restart the IDV and load in some weather text.
11. Type in some text to highlight and hit enter. Change the color. Does it change now?
12. Save this as your default bundle and then reload it. Notice that the text is not hightlighted this time. We need to add some bean property methods to save the selected text. Add the following methods:

```
/**
 * Get the highlighted text
 *
 * @return the highlighted text
 */
public String getHighlightedText() {
    return highlightedText;
}

/**
 * Set the text to hightlight
 *
 * @param text the text to hightlight
 */
public void setHighlightedText(String text) {
    highlightedText = text;
}
```

13. Since we use the FLAG_COLOR attribute any color changes we make get saved automatically.

Now, let's add in a point that shows where the station is. Remember, we passed along the station as a property to the data source. We can add those properties to the DataChoice by passing them along when we create the DirectDataChoice.

1. Edit the WxTextDataSource and where we create the DirectDataChoice, change:

   ```
                                   DataChoice.NULL_PROPERTIES);  //no properties
   ```

   to:

   ```
                                   getProperties());
   ```
2. Now, let's uncomment the section of code in the `WxTextDisplayControl.init` that adds in a displayable to the default ViewManager.
3. Compile the files.
4. Start the ExampleIdv and remove the default bundle:

   idve -cleardefault
5. Load in some weather text.
6. Change the color. What happens to the selector point?

5.7 Miscellaneous

5.7.0 IDV Persistence with XmlEncoder
The IDV provides a Java object persistence mechanism.

5.7.0 IDV Persistence with XmlEncoder

The IDV provides a Java object persistence mechanism.
The persistence mechanism is used to save bundles, colortables, station models, etc. The persistence is done with the ucar.unidata.xml.XmlEncoder class. The IDV creates the XmlEncoder with the:

```
    protected XmlEncoder getEncoder(boolean forRead) {...}
```

method. If you extend the IntegratedDataViewer class you could override this method to initialize the XmlEncoder differently.

If you create new classes that will be persisted there are some conventions your class needs to follow.

See an example bundle file here: example.xidv

- A public parameterless constructor - the "bean" constructor.
- Any attributes that are to be saved have public setters/getters of the form:

  ```
  public void set<Parameter Name>(type param)
  public type get<Parameter Name>()
  ```
- The XmlEncoder will look at and encode non-static public members so its a good idea not to have any.
- The parameter types can be most anything, primitive, arrays, lists, hashtables or an object of a class that is also persistable. While public static inner classes are handled, non-static inner classes will no be persistable.
- If an object is not persistable the XmlEncoder prints a warning and tries to degrade gracefully.
- Once we make a release of the IDV we try very (very, very) hard to always have new changes to classes be backwards compatible with any encoded version of the class. Bundles that were written 4 years can still (for the most part) be read back into the IDV.
- Though we have a mechanism to handle class package and name changes we try not to do this. If a public set method is removed (or the type is changed) the encoder flags that as a warning. If you want to remove or change an attribute you need to keep around the old set method and handle the value appropriately.

XmlDelegate

For classes that cannot follow the above conventions one can define a XmlDelegate for the class that handles the persistence for objects of that class.

```
public interface XmlDelegate {
    /**
     *  Create the xml element for the given object.
     *
     *  @param encoder The XmlEncoder that is calling this method.
     *  @param object The Object to encode.
     *  @return The xml element that defines the given object.
     */
    public Element createElement(XmlEncoder encoder, Object object);

    /**
     *  Create the Object defined by the given xml element.
```

```
     *  @param encoder The XmlEncoder that is calling this method.
     *  @param element The xml that defines the object.
     *  @return The Object defined by the xml.
     */
    public Object createObject(XmlEncoder encoder, Element element);
}
```

Most of the uses of this interface are derived from the XmlDelegateImpl which provides a default createObject method. The delegates overwrite createElement typically by putting on a *constructor* tag. For example, here we have a delegate that handles the persistence of Color objects:

```
XmlDelegate xmlDelegate =  new XmlDelegateImpl() {
         public Element createElement(XmlEncoder e, Object o) {
             Color color = (Color) o;
             List args = Misc.newList(new Integer(color.getRed()),
                                      new Integer(color.getGreen()),
                                      new Integer(color.getBlue()));
             List types = Misc.newList(Integer.TYPE, Integer.TYPE,
                                       Integer.TYPE);
             return e.createObjectConstructorElement(o, args, types);
         }
    });
```

The utility method

```
XmlEncoder.createObjectConstructorElement(Object o, List args, List types);
```

creates a *constructor* tag e.g.:

```
                  <object class="java.awt.Color">
                      <constructor>
                         <int>0</int>
                         <int>0</int>
                         <int>0</int>
                      </constructor>
                 </object>
```

To add a delegate to an XmlEncoder call:

```
XmlEncoder.addDelegateForClass(Class theClass, XmlDelegate xmlDelegate);
```

The XmlEncoder that the IDV uses is initialized with its own addDefaultDelegates() and with the ucar.visad.VisADPersistence.init(XmlEncoder encoder) method.

Right now there is now way to add delegates through a plugin.

XmlPersistable

A class can also implement the interface XmlPersistable and handle its own persistence:

```
public interface XmlPersistable {
    /**
     *  Create the xml representation of the object.
     *
```

```
     *  @param encoder The encoder.
     *  @return The xml representation.
     */
    public Element createElement(XmlEncoder encoder);


    /**
     *  Initialize this object from the given xml element.
     *  @param encoder The encoder.
     *  @param element The xml element representing this object.
     *  @return Return  true if it is ok to do the default processing for this node.
     */
    public boolean initFromXml(XmlEncoder encoder, Element element);
}
```

In this case it still needs a parameter-less constructor but it will handle its own persistence.

1.1 IDV Features

The IDV has many features that make it an ideal tool for analysis and display of geoscience data.

- **Integrated displays of a variety of data types** Details
- **Support for a variety of data access methods** Details
- **Multiple 2- and 3-D display types** Details
- **Interactive probes** Details
- **User defined formulas** Details
- **Easy configuration** Details
- **Bundling of user preferences** Details
- **Integrated documentation** Details
- **Extensive use of network resources:**
    - **Client/Server data access** Details
    - **Use of Java Web Start** Details

Footnotes:

Integrated displays of a variety of data types

The IDV supports combining displays of a wide range of geoscience data including satellite imagery, gridded data, surface observations), upper air soundings, flight tracks, NWS NEXRAD Level II and Level III RADAR data, NOAA National Profiler Network data and some forms of GIS data.

Support for a variety of data access methods

Directly reads netCDF, Vis5D, McIDAS AREA and map (OUTL*) files, NEXRAD Level II and III radar data, ESRI Shapefiles, and Arc/Info ASCIIGRID files. Provides access to other data formats (e.g., GINI, NEXRAD Level III, HDF) through the Abstract Data Distribution Environment (ADDE) and OPeNDAP/DODS protocols.

Multiple 2- and 3-D display types

Users can display the same data in a variety of 2- and 3-D visualizations such contours, color-shaded cross sections, isosurfaces and profiles.

Interactive probes

The IDV supports a variety of data probes that are used for creating location dependent value readouts and vertical profiles for multi-dimensional datasets.

User defined formulas

Supports user creation of derived products from a library of mathematical and meteorological operations.

Easy configuration

Much of the IDV, including the user interface, is configured through a set of eXtensible Markup Language (XML) files. Developers and users can easily tailor the IDV to provide custom applications based on the underlying framework by using their own set of configuration files. Everything from map projections to color tables to the available display components is configurable. A hierarchical set of XML files is used to control core, site and user preferences.

Bundling of user preferences

The state (loaded data sources and data depictions) of the IDV can be saved to an XML "bundles" at any time. These bundles can be used to pre-configure the application, or can be loaded interactively while the IDV is in use. Users can share saved data views, complete with annotations.

Integrated documentation

On-line User's Guide, Help Tips and a "Getting Started" guide are available from within the application.

Client/Server data access

Access to data from ADDE and OPeNDAP/DODS servers allows subsetting of large datasets. THREDDS catalogs of data holdings indexed in digital libraries (e.g. DLESE) can be used for discovery and usage metadata.

Integrated HTML Viewer

Web pages can be viewed in the special HTML viewer component. This viewer can interpret special embedded links to IDV displays, controls and IDV bundles within a standard HTML document, allowing educators to create class exercise in HTML, including powerful interactive, three-dimensional displays of remote data.

Use of Java Web Start

The IDV takes advantage of Java's Web Start technology to provide a seamless method for internet-based software distribution. The application can be launched from a web link pointing to a Java Network Launching Protocol (JNLP) file, which can include pointers to configuration files and bundles for customization. When a user clicks on the link, any JAR files or other ancillary data that has changed since the last time the application was launched are automatically downloaded and installed on the users system.

Collaboration features

Users can share IDV sessions locally in a classroom or remotely through the Internet. This facility is configurable as peer-to-peer or in a client-server mode. An incorporated chat facility and drawing tool aid in communication.

2.1.8 Starting the IDV with Java Web Start

You can also start the IDV through the Java Web Start program.

Using Java Web Start

Java Web Start is an application-deployment technology from Sun Microsystems that downloads all necessary software for an application and then starts the application with a single click in your Web browser.

One advantage of Web Start is that it will install the program on the first download and will automatically check for updates and retrieve new versions on subsequent start-ups. Web Start has the disadvantage that you have to first install Java (which includes the Web Start distribution).

1. For this workshop, you can launch the workshop version of the IDV through Web Start from:

   http://www.unidata.ucar.edu/software/idv/webstart/IDV/workshopIdv.jnlp

   The Java Network Launching Protocol (JNLP) file at that location specifies the location for the files to download. It also includes the `installplugin` specification that we used earlier.
2. After downloading the files, the IDV is automatically launched. You will be asked to accept a certificate of authenticity. Accept this and the IDV will start up.
3. Notice that this IDV looks the same as the one you started from the command line. In fact, it uses the same preferences that were used by the command line version, so starting either way should give the same results.
4. This was just a demonstration, so you can exit the IDV in preparation for the next section. For the rest of the workshop, we'll use the command line version.

For more about use of Web Start with the IDV see the IDV FAQ.

Mouse and Keyboard Use

This page summarizes use of the mouse and keyboard for controlling IDV displays using the default mouse and keyboard assignments. You can change the default behavior through the `Navigation` tab of the User Preferences accessed by the **Edit** **Preferences** menu.

Mouse Conventions

"Click" the mouse means to press and release a mouse button once.

"Double-click" means to click a mouse button twice quickly.

"Drag" means to press and hold down a mouse button, move the mouse and then release the mouse button.

All references to mouse buttons are in respect to a 3-button right-handed mouse:

- **MB1** (mouse button 1) is the **left** button.
- **MB2** (mouse button 2) is the **middle** button.
- **MB3** (mouse button 3) is the **right** button.

If you are using a left-handed mouse, MB1 is the right button and MB3 is the left-button.

If you are using a Mac with a single button mouse, MB3 is simulated using the **Option**+click combination. Not all functions listed below (e.g., panning) work with this combination on a one button mouse.

This table summarizes the operations you can perform with the mouse:

Mouse Button	Key Modifier	Purpose
MB1	none	Moving selector points in view windows, selecting items in lists and widgets, clicking buttons, moving sliders.
	Shift	Rubber band zoom in view window
MB3	none	Rotate 3-D displays
	Ctrl	Panning in view windows
	Shift	Zoom in/out

Zoom, Pan, and Rotate

There are several methods to zoom, pan, and rotate in the IDV. You can use the mouse, the arrow keys, a rubber band box, or the Zoom/Pan Toolbar.

Rubber Band Box to zoom in

You can zoom into an area by outlining it with a "rubber band box." Hold down the **Shift** key, then drag the left mouse button (MB1) to outline a box. The outline appears while you are dragging the mouse, before the zoom. If the display was significantly rotated before using the rubber band box, the new display may have the data area off screen; zoom out to find it, or use **Ctrl-r** key combination to reset to the initial overhead viewpoint.

Mouse controls to Zoom, Pan, and Rotate

You can zoom, rotate, and pan using the right mouse button (MB3).

- To **zoom** a 3D display hold down the Shift key and drag the right mouse button (MB3). Moving the mouse forward zooms in; moving the mouse backward zooms out. This method allows any amount of zooming and precise control.
- To **pan** a 3D display hold down the Ctrl key and drag the right mouse button (MB3) in the direction you want to move the display. This method allows the most precise control over how much panning occurs and allows panning in any direction.
- To **rotate** a 3D display in any direction, drag the mouse with right button down over the display. Rotating the display with the mouse takes some getting used to.

Keyboard controls to Zoom, Pan, and Rotate

You can zoom, rotate, and pan using the arrow keys on the keyboard. You must first click the mouse in the view window.

- You can **zoom** by holding the **Shift** key and pressing the up (zoom in) and down (zoom out) arrows on the keyboard.
- You can **pan** the display by holding the **Ctrl** key and pressing the arrow keys on the keyboard. Using the method, the display will pan in the direction of the arrow.
- You can **rotate** the 3D display using the arrow keys on the keyboard. Click in the window and use the arrow keys to rotate the 3-D box.
- Use the key combination **Ctrl-r** to reset the display to the initial view of this projection (overhead, centered on center of the map projection in use).

List selection

The IDV uses the standard methods for selecting multiple items in a list (e.g., times list). These are:

- **To select everything in the list:**:
    - Click on the list to activate it.
    - Press **Ctrl-a**.
- **To select a continuous range**:
    - Click on one item in the list to start the range.
    - Hold the **Shift** key down and click on the last item in the range. All items between the first and last are selected.
- **To select a non-continuous range**:
    - Click on one item in the list.
    - Hold the **Ctrl** key** down and click on other items in the list. As each item is clicked, it is highlighted to indicate that it is selected.
    - To unselected an item, hold the **Ctrl** key** down and click on the item.

On the Apple Mac, use the **Command key instead of **Ctrl**

Toggling Visibility of Displays

There are several ways to toggle visibility of displays.

1. When using the side legends, the `Display Visibility` checkbox to the left of the label can be used to toggle the visibility of that display. When using the bottom legend, click on the legend label to toggle visibility of that one display.
2. The padlock icon on the right side of the `legend bar` is a toggle to lock visibility. Click on the padlock icon to toggle locking. When locked, the icon will look like this. . This lock overrides most other actions which might change visibility, including using the keyboard number keys. Use this to lock one or more displays' visibility on or off while you change visibility of other displays. You can still toggle visibility of a locked display with a click on the legend label.

3. The function keys `F1, F2, F3` can be used to cycle visibility as follows:
   - The `F1` key cycles visibility through all non-locked displays, showing one at a time.
   - The `F2` turns on all non-locked displays.
   - The `F3` turns off all non-locked displays.

4. Use the **View Visible** checkbox menu item in each display control's menu.

Using the Map Selector

This page summarizes use of the map selector that is available in some of the choosers.
Some of the choosers (e.g., Level III radar, profiler, RAOB) use a map of stations. You can zoom and see more stations with a rubber-band box made by dragging the left mouse button on the map. Also note the icons below the map to zoom and pan. Reading from left to right, the plus icon zooms in, the minus icon zooms out, the curving arrow restores the previous view, and the house icon restores the original view. The up arrow moves the view up (map down); the down arrows moves the view down (map up), the left arrow moves the view left (map right) and the right arrow moves the view right (map left).

The Declutter check box allows you to show all stations (not checked), or only a limited number of stations that do not overlap each other (checked). You may need to zoom in to see all the stations clearly separated. TIP: It can be helpful to zoom into an area where you want to select stations, before clicking on the station, to avoid the chance of getting a neighbor station by mistake.

The station map may support selecting either one station (e.g. for radar data) or multiple stations (e.g., for profiler data). Left clicking on a station will select it. If in multiple selection mode Shift-drag will select all stations within a region. Control-drag will add the stations in the selected region to the set of already selected stations. Pressing the Control-A key will select all stations. Right clicking in the map will bring up a menu that allows you to select all stations and to clear the selection.

Starting the IDV using runIDV

Starting the IDV using the `runIDV` script.

1. Open a terminal window (if you do not have one open already) and change to the directory into which you installed the IDV. (e.g., `/home/idv//IDV_3.1`)

   ```
   cd   /home/idv//IDV_3.1
   ```
2. To start the IDV for the workshop, from that directory type:

   ```
   ./runIDV -sitepath http://www.unidata.ucar.edu/software/idv/data
   ```
3. If the `IDV Help Tips` window is displayed when you start the IDV and you don't want this to happen every time, uncheck the `Show this next time` box and click the `Close` button. You can always bring up the help tips by selecting the **Help** **Show Help Tips** menu.

IDV Local Datasets

Installing local datasets for the workshop materials

The IDV can access data stored on remote servers and the workshop materials try to take advantage of that where possible. Some data types (e.g., Level II Radar data) can only be loaded from local or network mounted files.

For those exercises that require data on a local or network drive, you will need to first install the data.

1. Download the workshop data from: ftp://ftp.unidata.ucar.edu/pub/idv/sample_data/workshopdata.zip
2. Once you have downloaded this file, unzip it on a drive/directory that will be accessible by the IDV.
3. For those sections of the workshop materials that ask you to load data from `/data/idv`, substitute the drive/directory where you unzipped the file.

Time Animation Widget

This page summarizes use of the Time Animation Widget for controlling animations in IDV displays.

Time Animation Widget

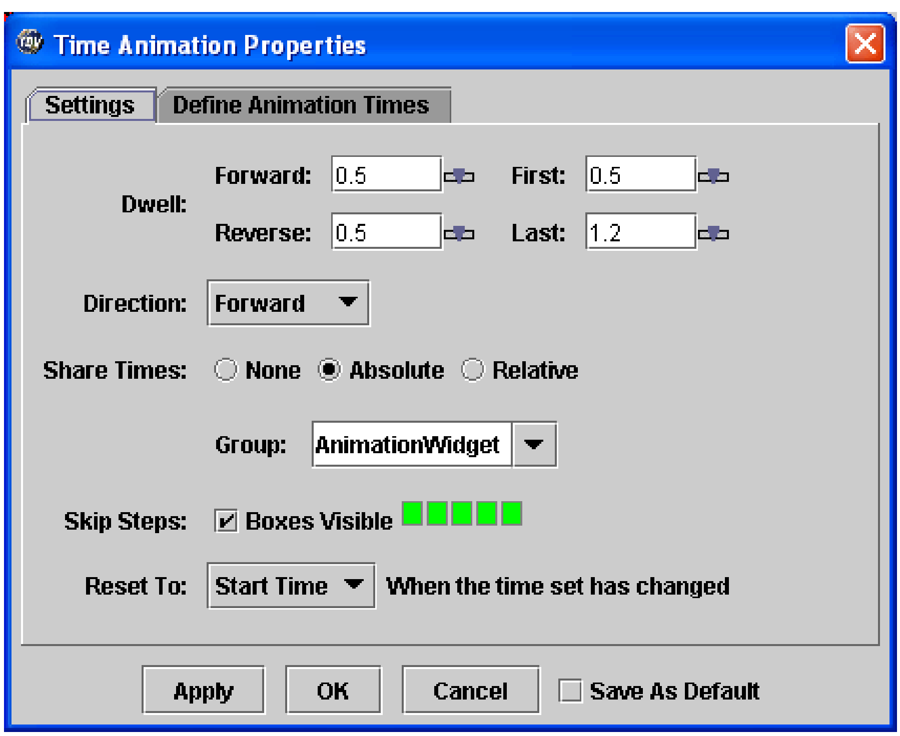
Each IDV view window has a set of time animation controls above the view window.

The `Time Animation Widget` controls looping of displays through time when more than one data time is loaded. The time of the data is displayed in the labeled pull-down list on the left. The toolbar of button icons, on the right, controls time animation. Each button works as follows:

- - Show the first frame, and stop looping if looping is on.
- - Step Back one frame.
- / - Toggle looping on or off. The icon switches between (start looping) and (stop looping).
- - Step Forward one frame.
- - Show the last frame, and stop looping if looping is on.
- - Show the Animation Properties dialog box.

Time Animation Properties Dialog

Click on the in the `Time Animation Widget` to see the `Time Animation Properties` editor.



The `Time Animation Properties` editor allows you to change the values that control looping. Drag the top four scroll bars to change their value. The properties include:

- `Forward Dwell` controls speed of looping forward (seconds/frame); dwell is how long each frame is seen when looping
- `Backward Dwell` controls speed of looping backwards (seconds/frame)
- `First Dwell` how long first frame is seen (seconds)
- `Last Dwell` how long last frame is seen (seconds)

You can set the loop Direction:

- `Forward` click this button to loop forward in time
- `Backward` click this button to loop backward in time
- `Rocking` click this button to enable rocking, where looping alternates between forward and backward directions.

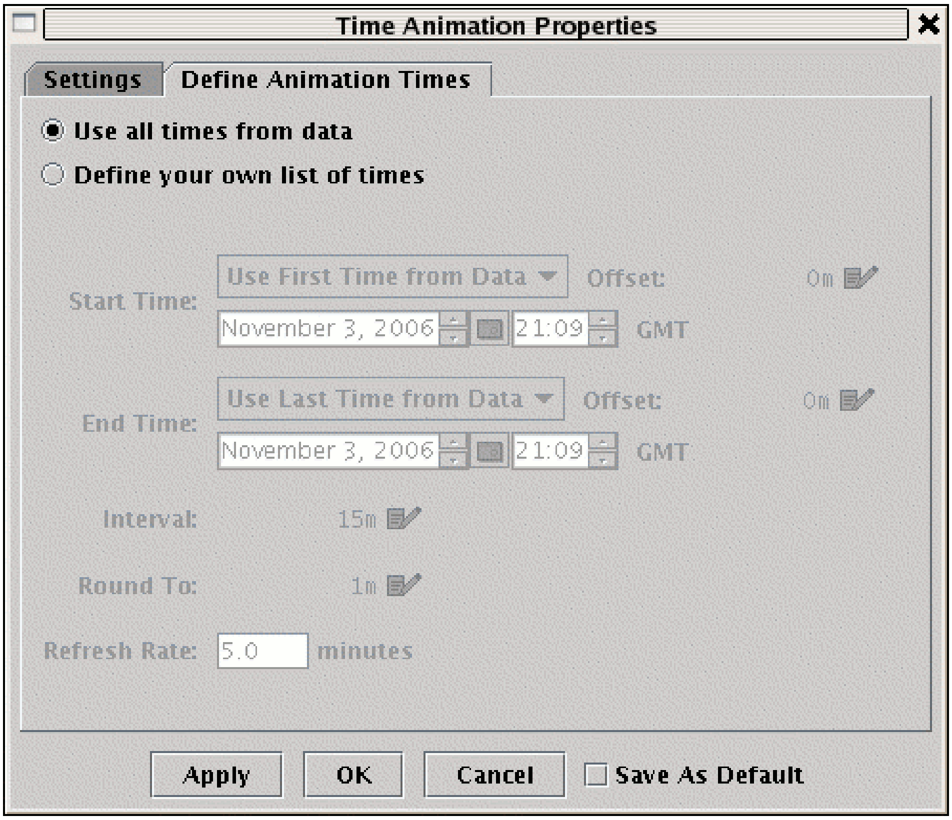
The `Share Times` options are used to toggle sharing of animations among displays. You can disable sharing `None`, share by `Absolute` time or by `Relative` indices between two displays. You can group different animation widgets and share among the group with the `Group` box. The `Boxes Visible` checkbox is used to show or hide the animation step boxes on the Time Animation Widget.

The `Skip Steps` boxes allow you to skip time steps in the sequence. Right click on a yellow box to skip that step and it will turn red. Right click on a red box and it will become yellow, indicating that the step is active.

The `Reset To` menu allows you to select what happens when new data/times are added to the display. e.g., reset to beginning time, reset to last time, keep at orginal time.

Defining Custom Animation Sets

Normally, all of the different times in a display are used to create the set of times that are looped through. However, the second tab in the Animation Properties dialog allows the user to define custom animation sets:



To define your own time set select the `Define your own list of times` button. The basic idea is that you define a start time, and end time and a time interval. These times can be set from the data, from a fixed time, the current time and relative to each other. An offset (both positive and negative) can be defined as well by clicking on the edit button: .

A simple case would be if you have a mix of satellite imagery (every 15 mintues) and radar (every 7 minutes). You could choose the start time and end time to be set from the data. And you interval could be 5 minutes. This would provide a uniform animation.

Another example is if you are displaying real time data (e.g., aircraft track) and you want to see the last hour at minute increments. In this case your end time would be "Current Time (Now)". You start time would be "Relative to End Time" with an offest of - 1 hour. Your interval would be 1 minute.

For real time you can define a refresh rate (e.g., every 5 minutes). This will automatically update the current time and rebuild the animation time set.

The `Round To` time can round the base times. e.g., to the nearest minute, hour, day, etc.

Using Jython in Layout Models

Parameter identifiers within a layout model can hold embedded Jython code. This allows you to do operations on the values and provide more extensive formatting.

1. Start fresh.
2. Open up the `Layout Model Editor` by selecting the **Tools Layout Model Editor** menu item.
3. Create a new Layout Model, called "Workshop2".
   - Select the **File New** menu item.
   - In the dialog box enter the name of the new Layout model, "Workshop2".
   - Click the `OK` button.
4. Add a `Value` symbol into the center of the display.
   - In the `Properties` window type in `T` in the `Parameter` field.
   - Click the `Save` button.
5. Load in some surface observation data.
6. Create a `Point Data` display.
7. In the `Point Data Plot` display control window make sure the layout model used is the new "Workshop2".
   - If it isn't click on the `Change` button and select "Workshop2" from the menu.
8. Back to the `Layout Model Editor`, change the parameter of the value symbol to be a calculation of dewpoint depression.
   - Right click on the value symbol to bring up the `Properties` window.
   - In the `Parameter` field type in:

     ```
     =T-TD
     ```
   - Any Parameter that begins with "=" is treated as a Jython expression.
   - Click `OK`
   - Save the `Workshop2` layout model with the **File Save** menu item.
   - You should see the `Point Data Plot` display change.
9. Now, let's add some formatting.
   - We want something that looks like " TD: <the value>"
   - Bring up the `Properties` dialog for the value symbol.
   - In the Parameter field type in:

     ```
     ='TD:'+(T-TD)
     ```

     This is Jython that returns the string 'TD:' concatenated (with the "+") the value of `T-TD`.
   - Click `Save`. Whoops, what went wrong? Look at the `Details` section.
   - This is a Jython error. We are trying to convert a double into a string value.
   - In the Parameter field change the line to:

     ```
     ='TD:'+str(T-TD)
     ```

     This changes the double value to a string.
   - Click `OK` and then save the changes by selecting **File Save**.
   - You should see the `Point Data Plot` display change.
   - Notice, though, that some of the values are rather long. This is because, normally, numeric values are formatted (e.g., stripping off extra decimals). However, because we are creating a

string in Jython there is no formatting occurring.

- To format repeat the steps above but enter:

  ```
  ='TD:'+format(T-TD)
  ```

  This formats the result of (T-TD) using the format descriptor for the symbol.

10. So, we are starting to get a bit cluttered with the embedded Jython. We can use the Jython editor to define a function that is callable by the value symbol.
    - Open the Jython library editor with the **Edit Formulas Jython Library** menu from the main menu bar.
    - Type in:

      ```
      def showdpd(t,dewpoint):
              dpd = t-dewpoint
              return 'TD:'+format(dpd)
      ```
    - Save this Jython with the **File Save** menu.
    - In the `Layout Model Editor` repeat the steps above for the parameter of the value symbol but enter:

      ```
      =showdpd(T,TD)
      ```
    - Click `Save`.
    - You should now see the new values in the `Point Data Plot` display.

Footnotes:
To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

To load in surface observation data:

- Open the `Surface Point Data Chooser` by either:
  - Select the `Point` tab in the `Data Chooser`.
  - Or, choose the **Data Choose Data Surface Observations** menu item from the main menu bar.
- Click on the `Connect` button.
- Make sure the `Data Type` is `Surface (METAR) Data`.
- Choose the latest time (the last in the list) in the `Available Times` list.
- Click the `Add Source` button.

To create a `Point Data` display:

- Select the `Surface (METAR) Data` in the `Data Sources` list in the `Field Selector`.
- Select `Point Data` in the `Fields` list.
- Select `Point Data Plot` in the `Displays` list.

3.8.3 Using the IDV Sounding Display

Using the IDV Sounding Display

The temperature and dew-point traces can be modified via the cursor. Press the left button on the first data-point to be modified and drag it horizontally along an isobar to the desired position, then move the pointer diagonally along an isotherm to keep the data-point at the desired position and to pick-up the next data-point. Repeat until done and then release the mouse button.

The check box labeled `Parcel mode` selects the origin of the path of a parcel of air lifted pseudo-adiabatically from the initial point to the top of the diagram. Check `Parcel path` to display the path of a lifted parcel.

The initial conditions of the pseudo-adiabatically lifted air-parcel (i.e. the parcel's pressure, temperature, and moisture content at the start of the pseudo-adiabatic lifting) are determined from the environmental sounding (i.e. the temperature and dew-point profiles) according to the mode indicated by `Parcel mode`. The modal choices are **Bottom of Sounding**, **Below Cursor**, **At Cursor Pressure**, and **At Cursor (Press, Temp)**.

In **Bottom of Sounding** mode, the initial conditions of the parcel are determined from the temperature and dew-point at the lowest (i.e. highest pressure) data-point of the environmental profiles.

In **Below Cursor** mode, the initial conditions of the parcel are determined from the temperature and dew-point profiles by computing the pressure-weighted mean potential temperature and water-vapor mixing-ratio of the atmosphere that lies below the cursor when the middle mouse button is pressed. Thus, pressing the middle button at 950 hPa for a sounding that starts at 1000 hPa will effectively mix the lower 50 hPa of the atmosphere and lift the resulting parcel starting at 975 hPa.

In **At Cursor Pressure** mode, the initial pressure, in-situ temperature, and moisture content of the parcel are taken from the the temperature and dew-point profiles at the pressure of the cursor when the middle mouse button is pressed.

In **At Cursor (Press, Temp)** mode, the initial pressure and in-situ temperature of the parcel equal the pressure and temperature values under the cursor when the middle mouse button is pressed. The initial moisture content is determined from the dew-point profile at the initial pressure.

The computed path of a lifted parcel is automatically updated when appropriate. In **Below Cursor**, **At Cursor Pressure**, and **At Cursor (press, temp)** modes, the path will update as the cursor is dragged. In all modes, the path will update whenever the profiles change.

The check box labeled `Virtual Temperature` controls whether or not the virtual temperatures of the environmental profile and pseudo-adiabatically lifted air-parcel are displayed.

The check box labeled `Spatial Loci` controls whether or not the location of the sounding data is displayed in the main, 3-D window. For aircraft data, the track of the aircraft will be displayed; for RAOB data, the locations of the stations will be displayed; and for gridded data, the grid points will be displayed.

The table to the left of the sounding diagram contains aerological parameters determined from the atmospheric sounding displayed in the diagram. The following abbreviations are used:

LCL
: Lifting Condensation Level: the level at which a pseudo-adiabatically lifted air-parcel becomes saturated with water-vapor.

LFC
: Level of Free Convection: the level at which the virtual temperature of a pseudo-adiabatically lifted air-parcel changes from being less than the environment's virtual temperature to being greater than the environment's.

LNB
: Level of Neutral Buoyancy: the level above the LFC at which the virtual temperature of a pseudo-adiabatically lifted air-parcel equals the environment's virtual temperature.

CAPE
: Convective Available Potential Energy: the area between the virtual temperature trace of the pseudo-adiabatically lifted air-parcel and the virtual temperature trace of the environment from the LFC to the LNB in which the parcel's virtual temperature is greater than the environment's (positive CAPE connotes convective development).

CIN
: Convective Inhibition: the negative of the area between the virtual temperature trace of the pseudo-adiabatically lifted air-parcel and the virtual temperature trace of the environment from the parcel's initial conditions to the LFC in which the parcel's virtual temperature is less than the environment's (negative CIN connotes initial positive work to lift the parcel).

3.9 Transect Views

3.9.0 Transect View

The transect view offers a 2D vertical transect.

1. Create a Transect and Map view window under `Misc` in the `New Window` tab in the Quicklinks. (Or from the menu: **File** **New** **View Window** **Misc** **Transect and Map**).
2. Close the original main Map View.
3. Load in the `Sample RUC Data` data source.
4. Click in the left Transect View to select it and create a Color Shaded Vertical Cross Section.
5. Change some of the state of the Transect:
    - Change the vertical scale with the **View** **Vertical Scale** menu.
    - Toggle on/off the Grid Lines and Display Scales in under the **View** **Show** menu.
    - Zoom and pan in the transect.
6. Make sure the left Transect View is selected. Select **3D Grid** **Derived** **Flow Vectors** in the Field Selector and create a Wind Barb Cross Section.
7. Where are we looking at?

Footnotes:
To load in the `Sample RUC Data` data source:

- Open the `Catalog Chooser` by either:
    - Choose the **Data** **Choose Data** **From a Catalog** menu item.
    - Click the `Catalogs` tab in the `Data Chooser`.
- Select the **Sample Data** **RUC Grid** item.
- Select the `Add Source` button.

3.9.1 Showing Transect Lines

Use the Transect Drawing Control to display and modify the transect lines.

1. Under the **Transects** menu select **Edit**.
2. The Transect Drawing Control is a modified drawing control that allows you to see what a Transect View is viewing and create and modify new transect lines.
3. The bright green line is the transect that the Transect View is showing.
4. Try zooming and panning in the Transect View and see what happens.
5. Change the projection in the Map View with **Projections Predefined Use Displayed Area**
6. Create some new transect lines.
    - From the `Controls` tab choose the `Create a Transect` mode .
    - Draw a number of transect lines in the Map View.
    - Select the `Shapes` tab and click on different rows and see what happens.
    - Double click on one of the transect rows and change its name and start/end labels.
7. Show multiple Transect Views.
    - Create another Transect View with **File New View Window Transect View One Pane**
    - Change the transect it is using through its Transect menu.
    - In the Transect Drawing Control click off `Show Transects`
    - Remove this new view window.

3.9.2 Setting Transect View Bounds

1. Remove all data and displays.
2. Recreate the Transect Drawing Control (**Transects Edit** menu).
3. Select the left Transect View.
4. Load in some surface observation data.
5. In the Point Data Plot turn off decluttering.
6. Whoa! That's not good.
7. The Transect View shows everything on the surface of the earth that falls within the lines that are orthogonal to the end points.
8. Let's change that.
    - Show the Transect View properties dialog with the **View Properties** menu.
    - Under the Transect tab change the Max distance to 30 KM and hit `Apply`.
    - In the Map View drag the transect line around.
    - Create a regular point data plot in the Map View

Footnotes:
To load in surface observation data:

- Open the `Surface Point Data Chooser` by either:
    - Select the `Point` tab in the `Data Chooser`.
    - Or, choose the **Data Choose Data Surface Observations** menu item from the main menu bar.
- Click on the `Connect` button.
- Make sure the `Data Type` is `Surface (METAR) Data`.
- Choose the latest time (the last in the list) in the `Available Times` list.
- Click the `Add Source` button.

3.9.3 DEMs in the Transect View

Let's use the Transect View to show elevations.

1. If other displays and data are already loaded, select the **Edit Remove All Displays and Data** menu item or the icon in the toolbar to clear them out.
2. Create a Transect and Map view under `Misc` in the `New Window` tab in the Quicklinks. (Or from the menu: **File New View Window Misc Transect and Map**).
3. Remove the existing Map View window.
4. We're going to load in one DEM file.
   - From the Files Data Source Chooser go to /data/idv/trex
   - Load the `sierrasdem.nc`
5. Select the Map View on the right by clicking in the window.
6. Select the sierrasdem data source in the Field Selector and create a Color-Shaded Plan View. **Note: for performance reasons do not try to create a Contour Plan View**
7. Select the Transect View on the left by clicking in the window.
8. Select the sierrasdem data source in the Field Selector and create a Data Transect display.
9. Add the Transect Drawing control with the **Transect Edit** menu.
10. Save this as a Favorite: `Dem in Transect` for a later exercise. (**Displays Favorites Save as Favorite**).

3.11.4 Shapefile

The IDV can display ESRI polygon shapefiles.

1. Clear all data and displays.
2. Load a shapefile. We'll use a catalog of shapefiles derived from the Census Tiger database provided by ESRI.
    - Open the `Data Source Chooser` window.
    - Click on the `Catalogs` tab.
    - Go to the `Main IDV Catalog`
    - Select `Tiger Shapefiles->Colorado->Roads->Boulder`
    - Press `Add Source`
    - The shapefile will be loaded and a `Shapefile Display` control will automatically be created.
3. Add a background image with the **Displays Maps and Backgrounds Add Background Image** menu item.
4. Note the shapefile display is hidden. Raise it with the `Vertical Position` slider.
5. This shapefile has an attribute database shown in the `Table` tab.
    - Select the `Table` tab.
    - Press the `Select Fields to Show` button to subset the fields.
    - Select all `Current Fields` by clicking in the list and pressing Control-A.
    - Press the `Remove` button.
    - In `All Fields` select `FENAME` (feature name)
    - Press the `Add` button.
    - Press `OK`
6. Let's just show Highway 119
    - Select the `Filters` tab.
    - Under `Property` column select `FENAME`
    - Select "Contains String"
    - Under `Value` column enter "119"
    - Press `Apply Filters`

3.11.6 WorldWind Locations

NASA'S WorldWind program comes with a tiled location database that allows for progressive disclosure of locations as you zoom in. The IDV makes use of this database.

1. Start fresh by clearing all displays and data and go to a CONUS projection (**Projections Predefined US CONUS** menu item).
2. Create the `WorldWind` display control from the **Displays Special WorldWind Locations** menu item.
3. This is just like the `Location` display control except it progressively discloses more detailed locations as you zoom.
4. Zoom into Colorado and watch how the display changes.
5. Zoom into Boulder. Remember, if things go blank you just zoomed past the surface of the Earth. Use the **Projections Use Displayed Area** menu item to reset the projection.
6. Add a `Background Image` and change the layer to `Urban Area`.
7. Keep zooming.
8. Remove the `Background Image` display control.
9. Change projections to some other part of the world and see what is listed.

Footnotes:
To start fresh choose the **Edit Remove All Displays and Data** menu item or the scissors icon in the toolbar.

IDV Workshop Conventions

Type face conventions used in this document:

Type face ***Description***

MenuMenus and menu items. For example, **File** **Default Bundle** **Save** would indicate that you click on the **File** menu, then the **Default Bundle** sub-menu and then the **Save** menu item.`Dialog`Windows and dialogs, including control windows`Parameter`Parameter and list names`Label`Used for denoting widget labels and titles`Display`Display types`Filename`File names`Dataset`Data set names`Link`Non-active Web link name describing an active link in another Web page

Regular Expression Patterns

This section provides a brief overview of regular expressions. For more in depth coverage there are excellent book (e.g., http://www.oreilly.com/catalog/regex/) resources.

Here are some example patterns and a description of what they will match:

Pattern	Description
some string	Will match match any occurrence of:"some string". e.g.: "Here is some string" will match. "Here is some string in the middle of the text" will match. "some string at the beginning" will match.
^*some string*	Will match "some string" at the beginning of the text: "Here is some string" will not match. "some string at the beginning"will match
some string$	Will match "some string" at the end of the text: "Here is some string" will match. "some string at the beginning" will not match.
^*some string$*	Will only match exactly "some string"
*a**	Will match 0 or more occurrences of "a"
a+	Will match 1 or more occurrences of "a"
.	Will match any one character.
.*	Will match 0 or more of anything.
.+	Will match 1 or more of anything.

4.4.2 IDV Property and Resource Files

Where does the IDV look for configuration information.

- At its heart, the IDV is configured by one or more properties files and a set of other `resource` files. By default the IDV looks in 3 different places:
  - System resource path - in the java jar files.
  - Site path - defined at the command line or through user preferences.
  - User path - e.g., /home/idv/.unidata/idv/DefaultIdv
- The property files define:
  - The application name.
  - What remote data servers are available.
  - What data catalogs to use.
  - Where to find the resource description files.
  - Etc.
- The workshop is using an idv.properties file that contains:

```
idv.title = Unidata Workshop IDV

##Semicolon delimited list of urls pointing to thredds catalogs
idv.cataloglist= /ucar/unidata/apps/workshop/workshopCatalog.xml;http://www.unidata.ucar.e

## default projection
idv.projection.default=World
```

- We are configuring the list of ADDE servers and their image and radar descriptors, and the `workshopCatalog` file for the gridded data chooser.
- The default start up process is as follows:
  - Process the default system property file.
  - Process any command line properties files.
  - Look in the properties loaded so far to get and process a list of other property files. These are:
    - <Site path>/idv.properties
    - <User path>/idv.properties
  - Look for the *idv.resourcefiles* property. This defines where to find the resource description files. e.g.:

    ```
    %USERPATH%/idv.rbi;%SITEPATH%/idv.rbi;%IDVPATH%/idv.rbi
    ```
  - Load in plugins (more on this later).
- Now we'll look at how the IDV finds the resources using the *rbi* files.

4.4.3 RBI File

The RBI file specifies where everything is.

- The RBI (Resource Bundle for the IDV) file is where the real work of configuration is defined. These are XML files that specify where to find all of the resources that the IDV uses to configure itself.
    - Jython libraries.
    - Color tables.
    - Projections.
    - Etc.
- As a convenience, the IDV places an example resource file in your local directory: idv.rbi
- Here is the RBI file that the IDV comes with: idvfull.rbi
- Lets look at how the IDV uses these resources. From the command line run the IDV with the *-listresources* argument:

```
idv  -listresources
```

- This will show a dialog that lists all of the resources specified by the RBI file (or files).
- Let's go and change the the local idv.rbi file in ~/.unidata/idv/DefaultIdv and add some resource paths to the color tables entry. e.g.:

```
  <resources name="idv.resource.colortables" loadmore="true">
    <resource location="some location"/>
    <resource location="some other location"/>
    <resource location="http://www.unidata.ucar.edu/software/idv/colortables.xml"/>
  </resources>
```

- Now lets run the IDV again and look at the color tables entry that we just changed.

```
idv  -listresources
```

4.4.4 Resources

So, how do you put all of this together?

1. Remember all of those resources we created in earlier exercises? They are saved off in the user's local resource directory: `/home/idv/.unidata/idv/DefaultIdv`.
2. The files include:
   - `aliases.xml` - The data aliases.
   - `default.xidv` - The default bundle.
   - `main.xml` - User preferences.
   - `paramdefaults.xml` - Parameter defaults.
   - `projections.xml` - Projections.
3. The IDV looks in the user's resource directory, the site path and the system path for these resources.
4. View the available resources
   - From the **Tools** menu select Resource Manager.
   - This provides an overview of all of the available individual resources.
   - But where are these individual resources kept?
     Answer...
   - To see just where the resources are located select the `List Sources` button.
5. To share these resources across a site one just needs to copy these files to an accessible web site or directory. Then point the IDV to that path with the *-sitepath* command line argument. Try running the IDV with:

   ```
   idv -userpath ~/otheridv -noplugins -listresources -sitepath http://www.unidata.ucar.edu/s
   ```
6. Note: when running with the -sitepath argument that value is persistent. Bring up the Preferences and remove the sitepath entry.
7. Exit the IDV.
8. Ahh, you might say: "what about these plugins I keep hearing about?"

4.4.5 Configuring using Web Start

Using Web Start JNLP files to configure the IDV.

1. Using Web Start to pre-define plugins.
   - You can create your own Web Start files that contain site configuration information.
   - Look here to view an example `JNLP` file.
   - You can pass in command line arguments. e.g.:

     ```
     <argument>-plugin</argument>
     <argument>http://www.unidata.ucar.edu/software/idv/data/myplugin.jar</argument>
     ```

   - You can also define bundles that are loaded at start up time:

     ```
     <argument>http://www.unidata.ucar.edu/software/idv/data/example.xidv</argument>
     ```

2. You can also configure the IDV to only have one running instance.
   - In the jnlp file add the *-oneinstanceport* argument and provide some port number.

     ```
     <argument>-oneinstanceport</argument>
     <argument>8888</argument>
     ```
   - Now, once the IDV is running with this argument subsequent executions of the IDV will result in the command line arguments being passed to the initial IDV and the new IDV exits.