

OpenDx Instruction for Color2D.net

© 2008 R.W. Clark

<http://www.cyberanalysis.com>

Table of Contents

OpenDx Instruction for Color2D.net.....	1
Introduction.....	1
Data Explorer – opening the application.....	2
The File Dialog Window – a mystery of complexity.....	2
Color2D Net – opening a representative file.....	3
Visual Program Editor – looking under the hood.....	4
The Image Tool – the presentation layer.....	5
The Collect Tool – combining image elements.....	6
The Color Tool – how colors are named and valued.....	7
The Colormap tool – putting color into the data image.....	8
The ColorBar Tool – putting a scale into the Image presentation.....	9
The Import Tool – connecting to the data.....	10
The Colormap Editor – creating a mapping of color to data.....	11
Appendix A – data metafiles and data files.....	12
Appendix B – a list of coloring names and values.....	13

Introduction

OpenDX is a data visualization tool with many facets of control over presentation. In fact, so many that it can make this package rather difficult to approach. This paper will, by degrees, cover the same ground offered in their tutorial, but I judge that attempt to be written from the perspective of those who already know the package and is written not so much as an introduction as it was written as an assignment, or as an obligation.

It plunges the reader immediately into the thick of arcane minutia without any overview or any means to grasp what is necessary to simply start with your own data and make a presentation. The problem is that OpenDX requires you to check the gas tank and look under the hood before you can get down the road. It also suffers from a horribly complex file dialog design. However, the effort of getting past these difficulties makes it worth the effort, and those difficulties are trivial once you gain experience.

Data Explorer – opening the application



Illustration 1: Data Explorer

Following the starting of the *Xwin server* to support *OpenDX*, then open Data Explorer. This will present a simple interface with a column of buttons. Select **Samples....**

Using the file dialog window that opens, traverse to the tutorial sub folder by using a combination of the file tree prompts and the **Filter** button. **Select** Color2D.net

The File Dialog Window – a mystery of complexity

This window's contents of panes and buttons presents a confusing assortment of controls that stumble to accomplish what the standard graphical interface requires for selecting and opening a file.

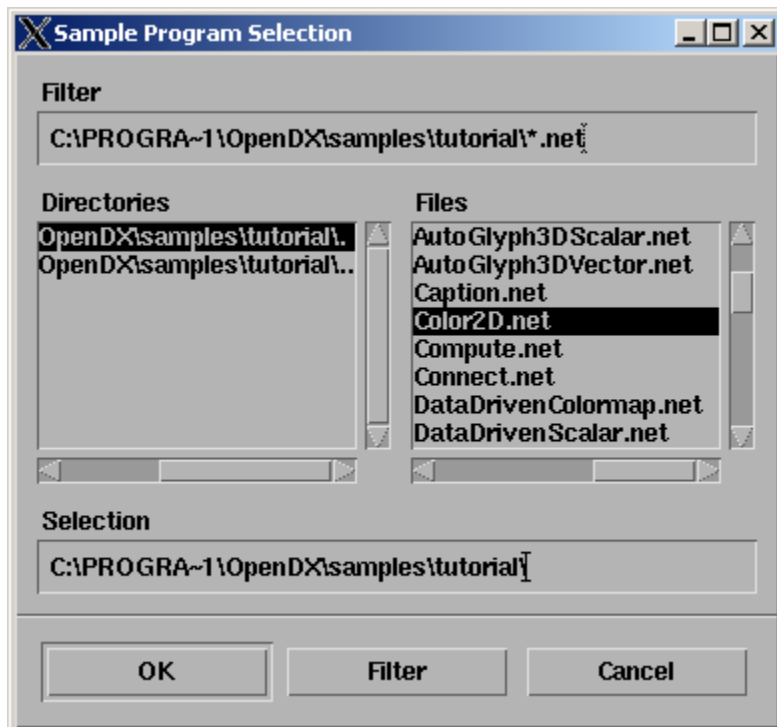


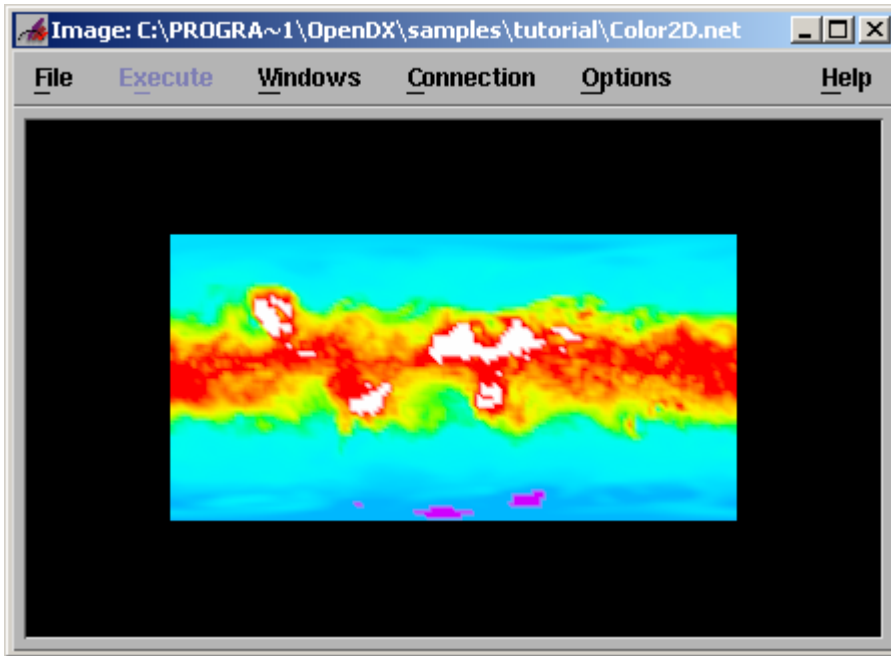
Illustration 2: The File Dialog Window

A Filter is a technical word for describing the target file in terms of its folder name (here it includes a rather more complete path) and filename extension (in this case *.net*). From these two the pane for **Directories** and **Files** is filled with candidates for the **Selection** box.

The method of traversing through folders to arrive at any particular file is rather cumbersome, but achievable through the maintenance of the combination of all these controls. To say more would be counterproductive to simply experimenting for yourself.

Color2D Net – opening a representative file

The following annotated screen shots are taken from OpenDX/samples/tutorial/Color2D.net. The picture offered here is a distribution of temperatures around the earth, where the equatorial hot region is denoted by the reds, where the temperate zones are in green and arctic regions in blue. This reveals the auto color feature of OpenDX with red being the highest values (hottest temperatures), and blues being the lowest (coolest temperatures).

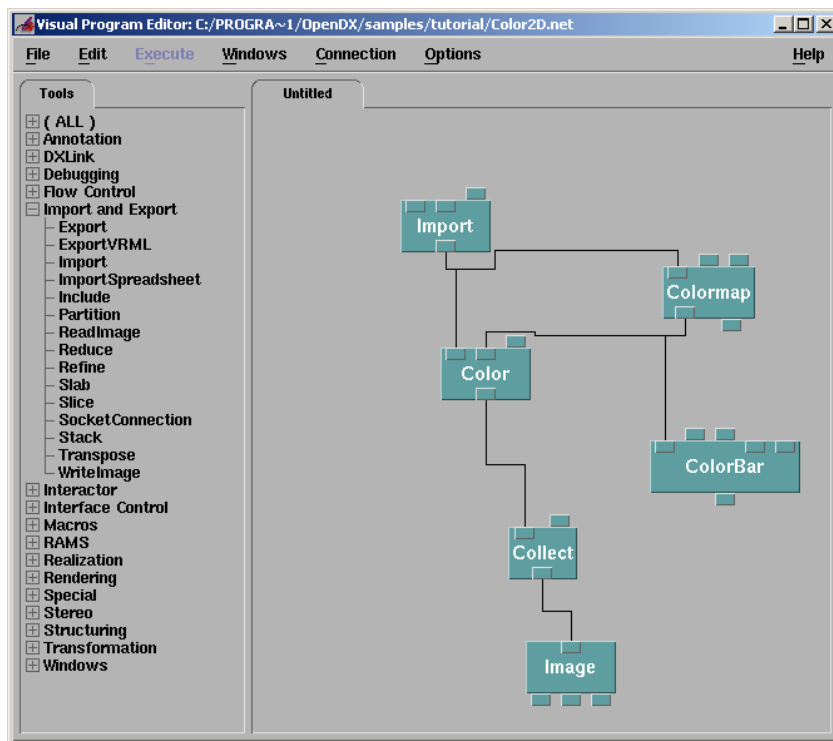


*Illustration 3: Data Explorer Image window which is the end process for viewing after the tool **Image** has been executed (seen in the Visual Program Editor window).*

*To view this tool, select the menu item **Windows – Open Visual Program Editor** from the drop down list.*

Visual Program Editor – looking under the hood

As suggested by the image opened earlier, we examine the outcome of opening its editor to reveal the components responsible for obtaining the data and managing its presentation.

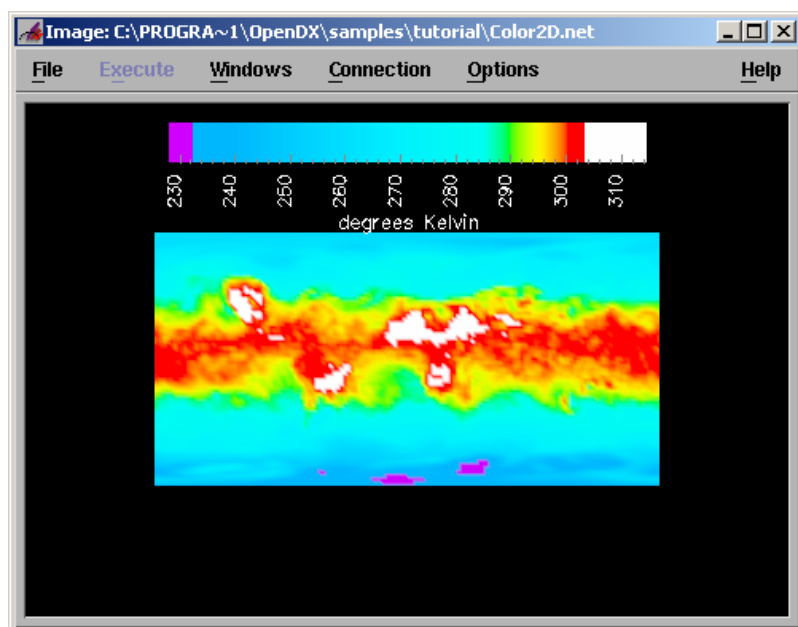


When using this, or any of the Editor tools, it is useful to select **Execute – Execute on Change** to see the immediate effects of any adjustments made to settings.

Here we are going to make a few changes in the existing design to observe how they affect the **Image** presentation.

Connect the output tab of the **ColorBar** icon in the Visual Program Editor window to the open input tab on the **Collect** icon.

Illustration 4: Visual Program Editor Opened from Data Explorer Image window. This original design will be edited so that the image can be altered.



When the change is made (not shown in Illustration 4) the Image presentation will follow with what is shown here in Illustration 5.

Illustration 5: Changed Image Presentation

The Image Tool – the presentation layer

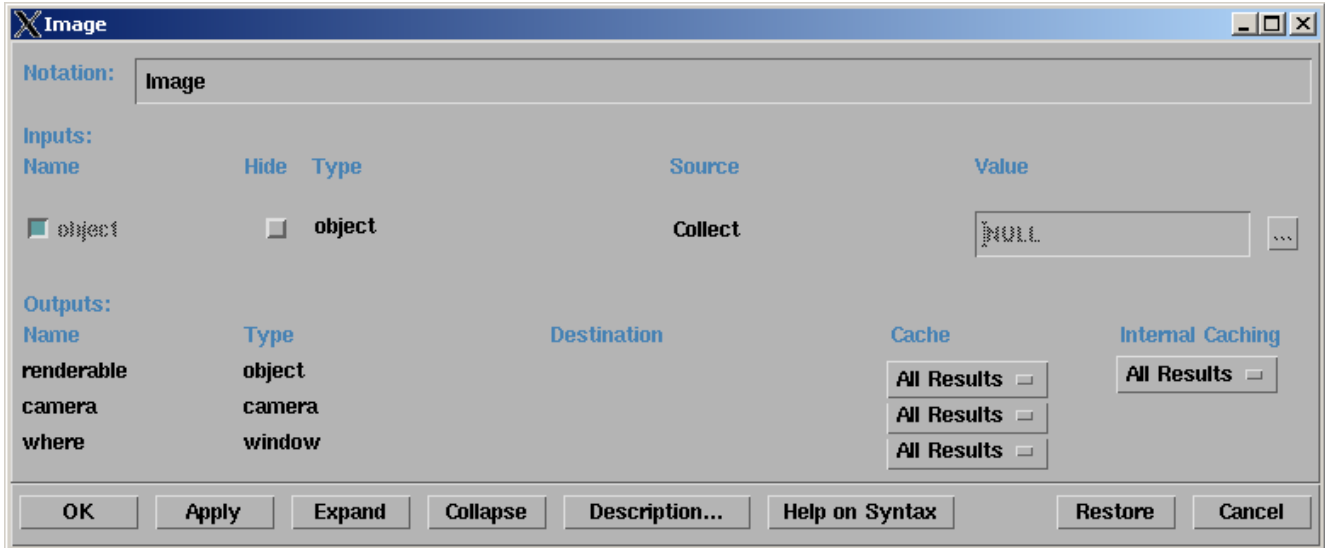


Illustration 6: The **Image** Tool Configuration.

This window is obtained by first selecting the **Image** tool in the Visual Editor, and then going to the Visual Editor menu bar and selecting **E**dit – **C**onfiguration.



Illustration 7: Image Tool Description...

The Collect Tool – combining image elements

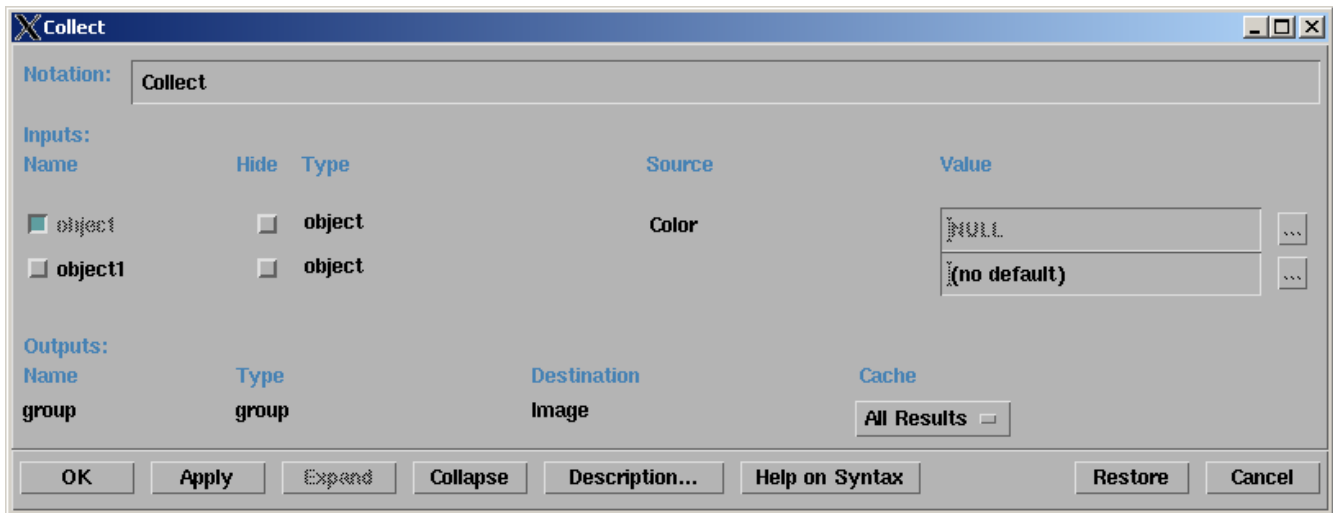


Illustration 8: The **Collect** Tool Configuration

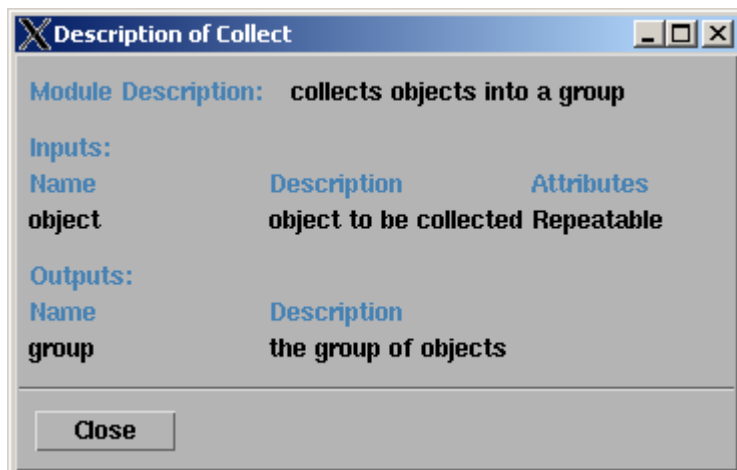


Illustration 9: Collect Tool Description...

The Color Tool – how colors are named and valued

The color tool supplies the settings of RGB for a text name for a color. There are three ways that this can be accomplished, with two as file look-ups. The file `color.txt` (see Appendix B – a list of coloring names and values) contains the look-up table and is located in one of two, or both places: in `/lib` or in `DXROOT/lib` where `DXROOT` is an environmental variable set by the user. Barring either of these two locations having the file, then a static table within OpenDX is used.

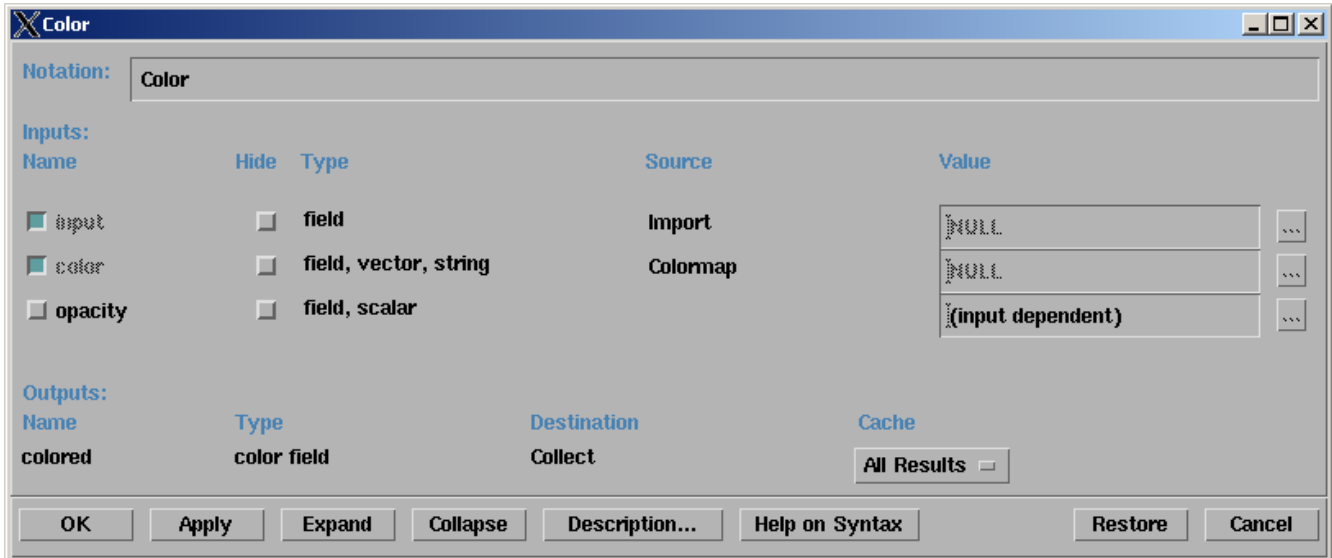


Illustration 10: The **Color** Tool Configuration

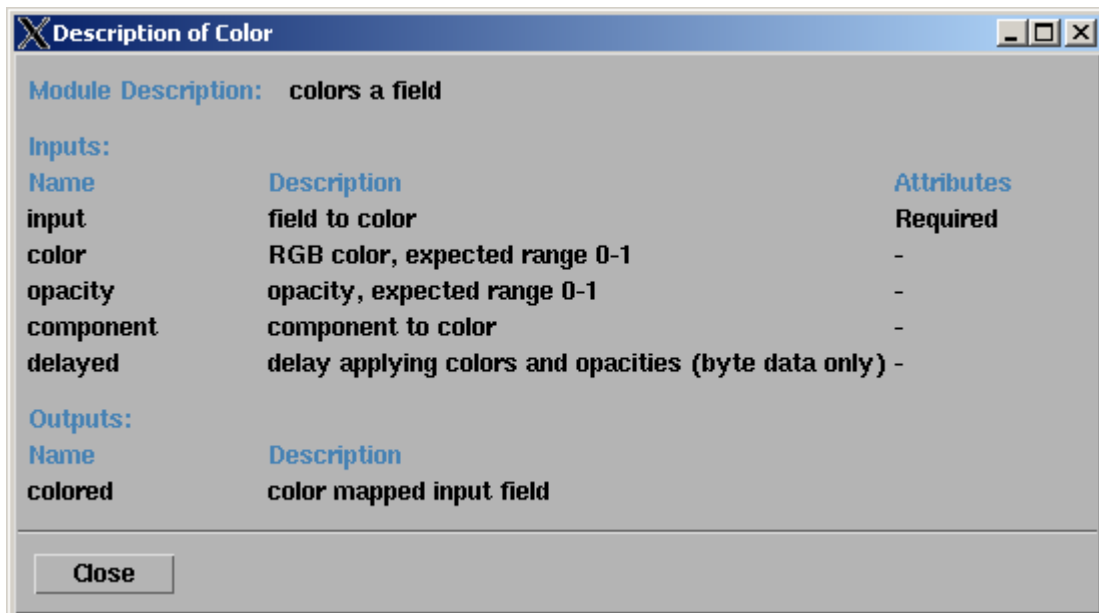


Illustration 11: The Description... Window for the Color Tool

The Colormap tool – putting color into the data image

The initial configuration for Color2D.net is shown here in Illustration 12.

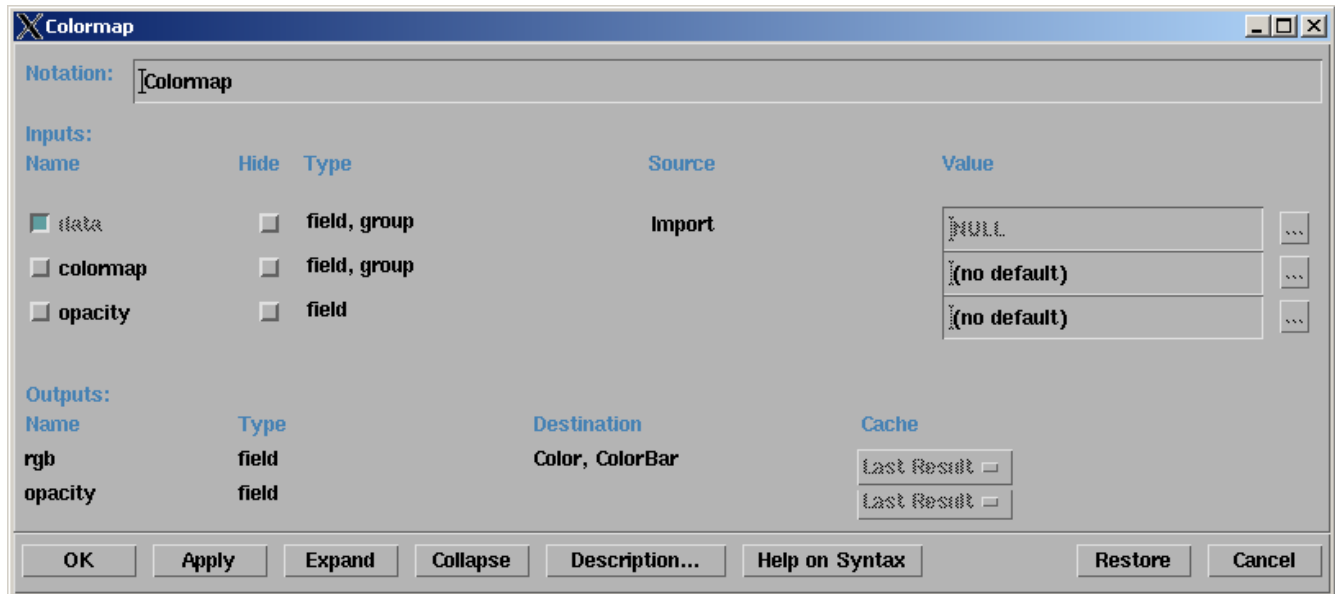


Illustration 12: The **Colormap** Tool Configuration.

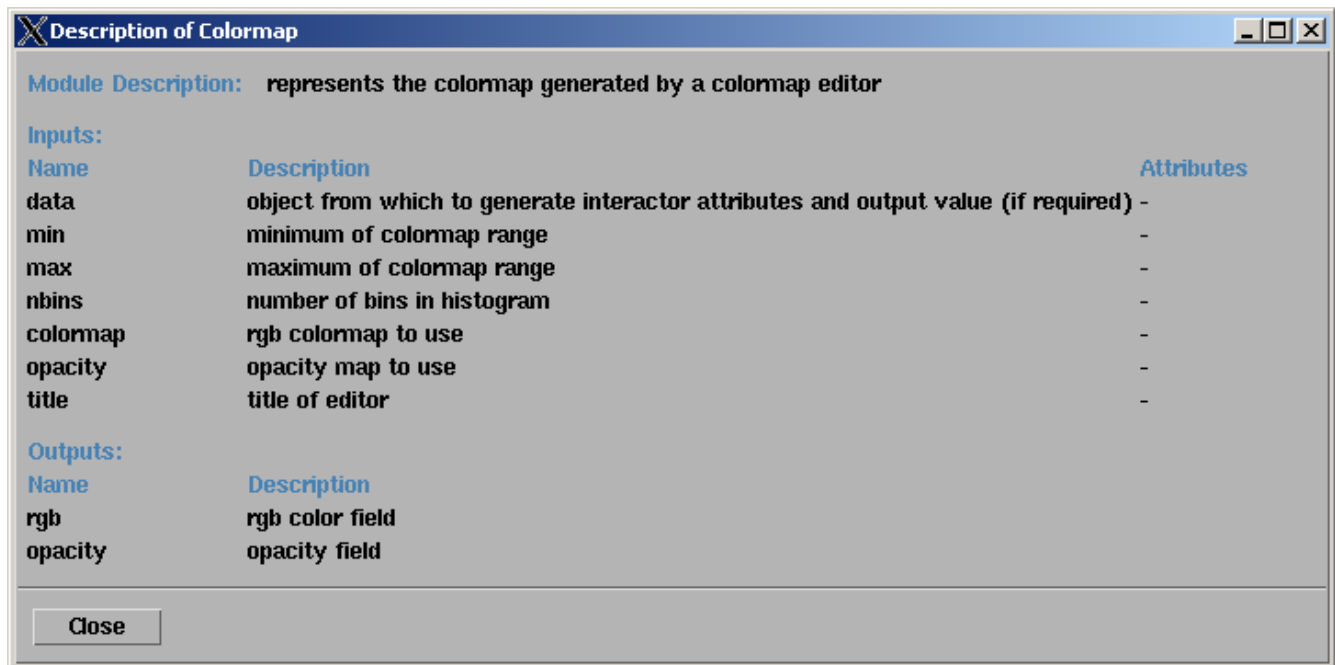


Illustration 13: The Description... for the Colormap Tool

The ColorBar Tool – putting a scale into the Image presentation

In the initial design of Color2D.net, this tool (shown in Illustration 14) resides with the other tools, but has no output, and thus makes no contribution to the Image itself. However, when we change the connections in the Visual Program Editor to connect it to the Image tool, as described above and in the IBM tutorial; then it provides a scale presentation within the Image. This scale correlates color to numeric data.

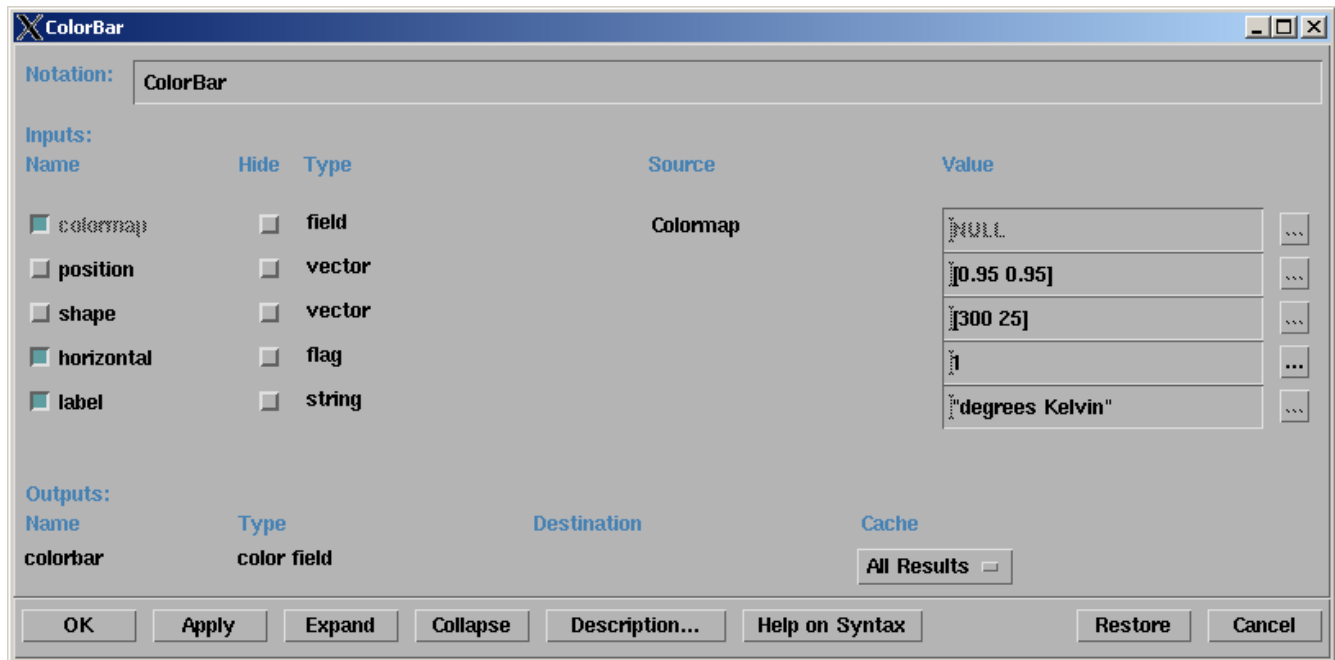


Illustration 14: The initial **ColorBar** Tool Configuration.

The Import Tool – connecting to the data

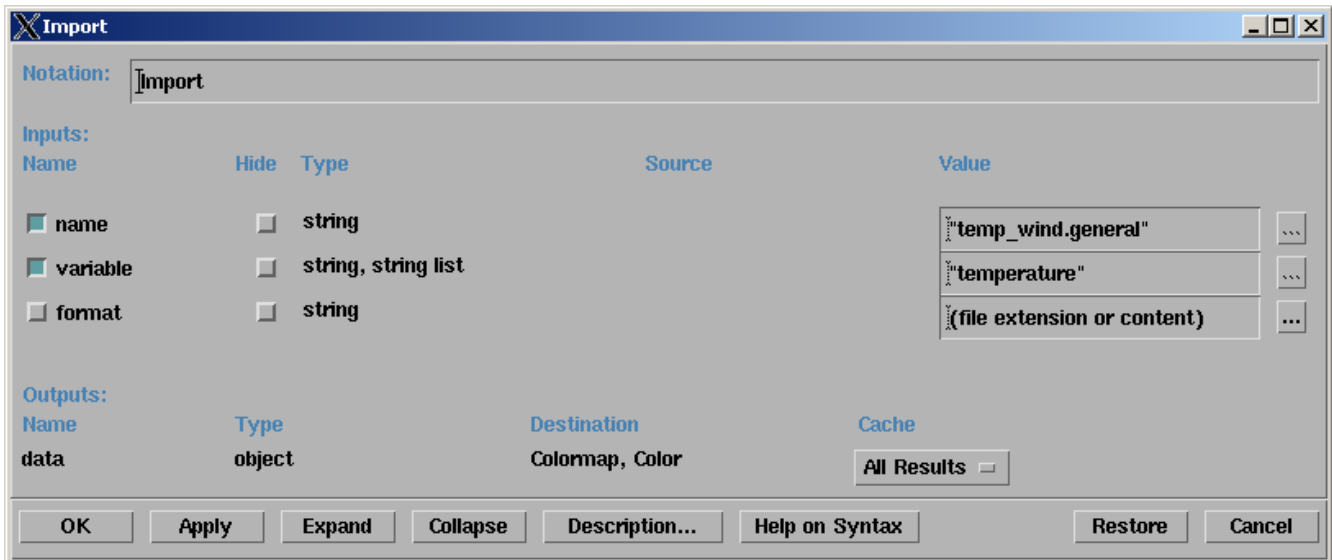


Illustration 15: The **Import** Tool Configuration.

The Import file name, `temp_wind,general`, is a configuration file in its own right that points to the data file `Temp_wind.Lis` (see appendix).

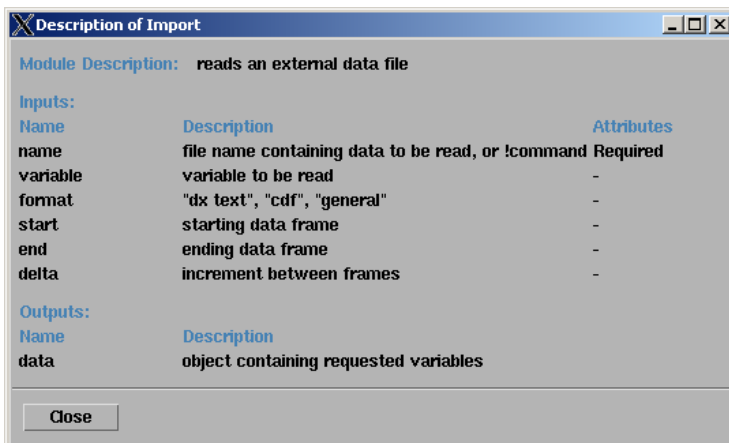


Illustration 16: The Description... window for the Import Tool

In Illustration 15, the variable selected called “temperature” is correlated in the file “temp_wind.general” to the field name of the same name, and the structure (type) of scalar.

The data file(s) are described in Appendix A. It should be noted that they are generated on the basis of earlier work employing the Import Data selection of the Data Explorer's interface shown in Illustration 1.

The Colormap Editor – creating a mapping of color to data

From the Image window, select **Windows – Open All Colormap Editors**. When using this, or any of the Editor tools, it is useful to select **Execute – Execute on Change** to see the immediate effects of any adjustments made to settings.

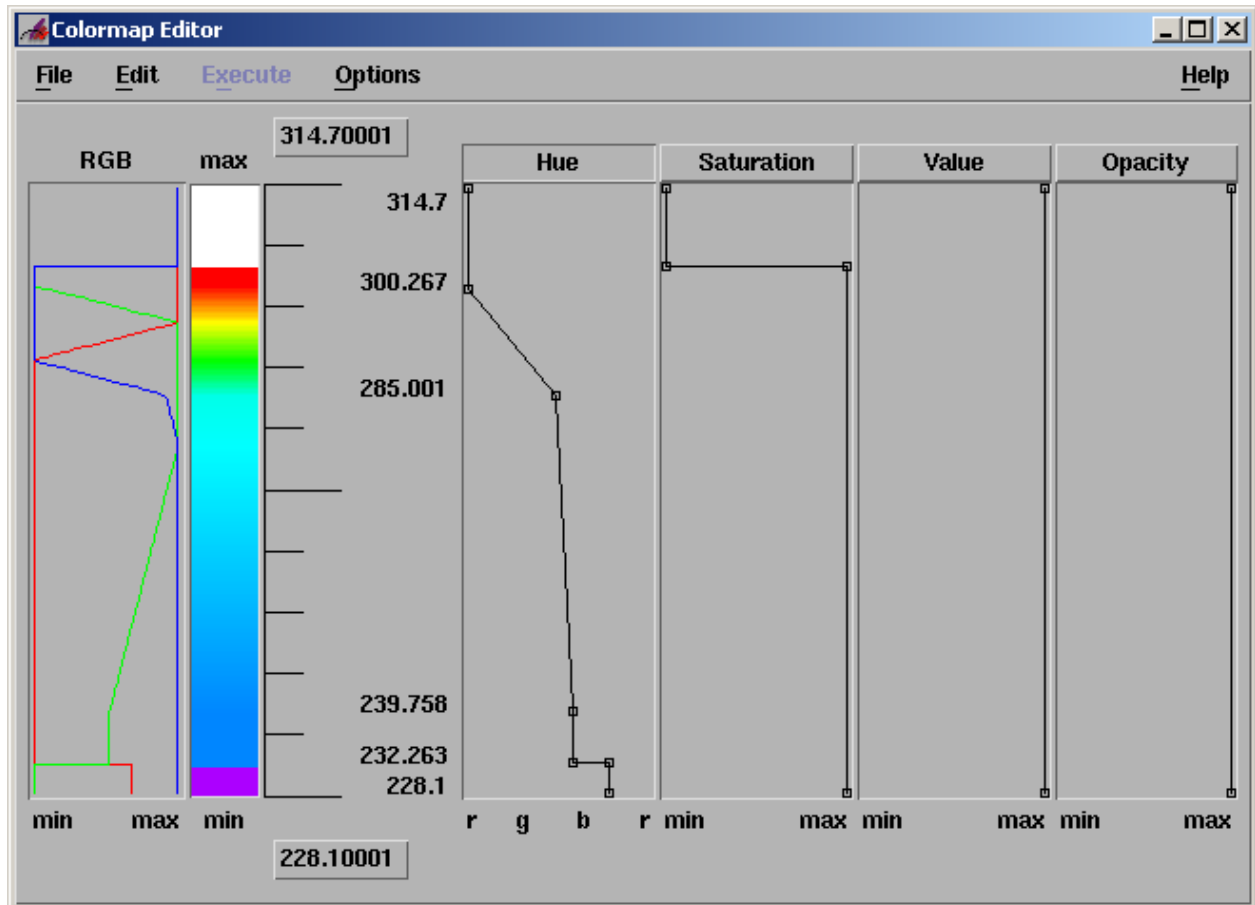


Illustration 17: The Colormap Editor

Appendix A – data metafiles and data files

FILE *temp_wind.general*:

```
# This is a general array importer header file which will import the data
# in "temp_wind.lis". This data is on a regular grid of longitude and latitude.
# There are 144 longitude points, and 73 latitude points. At each location there
# is a value for temperature (scalar) and wind velocity (2-vector).
# The layout keyword describes how the data are formatted. The first number
# is the number of characters to ignore before reading the first data value.
# The second number is the number of characters for the temperature. The
# third number is the number of characters to skip before finding the
# wind value, and in this case is 0. The fourth number is the number of
# characters for each component of the 2-vector for wind.
# The majority keyword indicates that the data is organized by column
# majority, meaning that the first dimension (longitude) varies fastest.
# That is, the data values for the first longitude and latitude position are
# read, then the data values for the second longitude and first latitude are
# read, and so on. The interleaving keyword indicates that the temperature
# and wind fields are interleaved. The header keyword indicates how many
# lines at the top of the file to skip before reading the data, and the
# positions keyword gives the longitude origin and delta, and the latitude
# origin and delta for the grid.
file = temp_wind.lis
grid = 144 x 73
field = temperature, wind_velocity
structure = scalar, 2-vector
# 25 bytes for time and 14 bytes for pressure (= skip 39 bytes as non-data)
# include 14 bytes for temperature scalar; skip 0 bytes and include 14 bytes for wind velocity 2-vector
layout = 39, 14, 0, 14
majority = column
interleaving = field
header = lines 8
positions = -178.75, 2.5, 90.0, -2.5
```

FILE *temp_wind.lis*:

VARIABLES AND SPECIFIED RANGES

```
-----
EPOCH          01-Oct-1987 00:00:00.000 01-Oct-1987 12:00:00.000
PRES_LVL       1000.000          1000.000
TMP            150.000          350.000
WIND_U        -100000000000.000 100000000000.000
WIND_V        -100000000000.000 100000000000.000
  EPOCH      PRES_LVL    TMP      WIND_U    WIND_V
01-Oct-1987 00:00:00.000 1000.000 258.500 6.500 13.200
01-Oct-1987 00:00:00.000 1000.000 258.400 6.700 12.600
01-Oct-1987 00:00:00.000 1000.000 258.300 6.900 12.000
```

Appendix B – a list of coloring names and values

FILE *Color.txt*

snow	1.0000000	0.9611688	0.9611688
ghostwhite	0.9458516	0.9458516	1.0000000
whitesmoke	0.9231065	0.9231065	0.9231065
gainsboro	0.7443291	0.7443291	0.7443291
floralwhite	1.0000000	0.9611688	0.8858132
oldlace	0.9843752	0.9231065	0.8135332
linen	0.9611688	0.8858132	0.8135332
antiquewhite	0.9611688	0.8492888	0.7108805
papayawhip	1.0000000	0.8784468	0.6977163
blanchedalmond	1.0000000	0.8492888	0.6462899
bisque	1.0000000	0.7994464	0.5907882
peachpuff	1.0000000	0.7308574	0.5263361
navajowhite	1.0000000	0.7579239	0.4602692
moccasin	1.0000000	0.7994464	0.5038216
cornsilk	1.0000000	0.9458516	0.7443291
ivory	1.0000000	1.0000000	0.8858132
lemonchiffon	1.0000000	0.9611688	0.6462899
seashell	1.0000000	0.9231065	0.8711111
honeydew	0.8858132	1.0000000	0.8858132
mintcream	0.9231065	1.0000000	0.9611688
azure	0.8858132	1.0000000	1.0000000
aliceblue	0.8858132	0.9458516	1.0000000
lavender	0.8135332	0.8135332	0.9611688
lavenderblush	1.0000000	0.8858132	0.9231065
mistyrose	1.0000000	0.7994464	0.7785468
white	1.0000000	1.0000000	1.0000000
black	0.0000000	0.0000000	0.0000000