

Leonardo 3.5

Michael Bruse

# **LEONARDOGraph**

Version from 03.05.2006

# 1 What is LEONARDGraph?

Graph is a LEONARDO Module to generate line series plot from different data sources. It integrates into the LEONARDO interface extending its visualization options from 2D data (maps) and 3D data (3Dmaps) towards high-quality line series charts.

LEONARDO Graph does not have an integrated charting module to create the diagrams, but uses the GNUPLOT ([www.gnuplot.info](http://www.gnuplot.info)) software engine to obtain high-quality outputs. Technically spoken, LEONARDO Graph is an interactive interface to the GNUPLOT software which facilitates the generation of plots to a huge extent.

You must not be familiar with GNUPLOT in order to use LEONARDO Graph, but if you are, you have the possibility to modify all operations using your own scripting commands by still profiting from the visual interface provided by LEONARDO.

LEONARDO Graph is designed to work with GNUPLOT Version 4.0 or newer. The basic GNUPLOT binaries are installed when you install LEONARDO. However, at any time you can replace them by newer GNUPLOT versions as soon as they become available. The minimum requirement for running LEONARDO Graph is GNUPLOT Version 4.0

## General approach

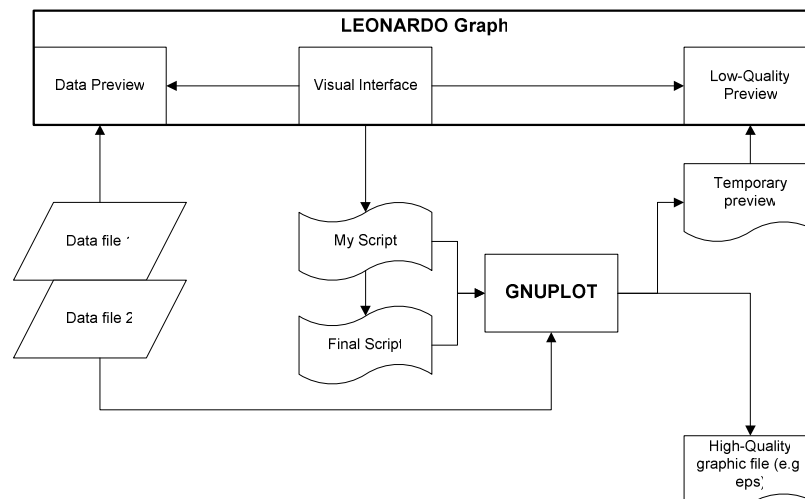


Fig 1. General architecture of LEONARDO Graph

Figure 1 shows the general data flow architecture of LEONARDO Graph as used in LEONARDO 3.5.

The program provides the following services in the preparation of your diagram:

- Preview of the data file(s) selected
- Interface to define different settings of the chart including the scripts
- Low quality preview of the final GNUPLOT chart

The general approach of LEONARDO Graph is to generate a GNUPLOT **Script** based on your settings and data file. This script will then be passed to

the GNUPLOT software, where a graphic file is created (you don't see GNUPLOT working... this all happens in a few milliseconds...)

There are two scripts LEONARDO Graph works with:

**My Script** – which is YOUR script which will be used to create the **final, high quality graphic file**, and

**Final Script** which is generated by LEONARDO and forwarded GNUPLOT in order to generate the chart.

You will only modify My Script- because this is your working script which will not be touched by LEONARDO Graph. Making changes in the Final Script won't have any influence because it is automatically generated by LEONARDO

## ***Basic steps when creating charts***

There are a number of differences between the way LEONARDO Graph works with data and graphs and how other software, say e.g. Microsoft Excel does it. The most important of all is that there is a **strict separation** between **data** and **visual layout** settings. LEONARDO Graph creates **scripts** for generating charts out of existing data- it does **not allow to modify the data**. When you load a data file into the interface, it is displayed for visual purposes only, so that you know how the data look like and that you can easily determine which data is in which row- you cannot edit the data.

After loading the first data file, LEONARDO Graph analyses the structure of the file and tries to display the names of the columns in an appropriate way. The way, the files are interpreted is similar to the way GNUPLOT interprets the files, e.g. in terms of header structure. If you are using the ENVI-met model, you will find pre-defined filter for the typical ENVI-met 1D files such as the inflow profile or different kind of preceptor files.

There are typically 3 steps involved in producing a chart diagram using LEONARDO Graph, which will be presented next.

### **1. Step: Compile and preview the diagram**

Once you have loaded your data file(s), you need to define the typical parameters needed to plot a chart diagram (define the x-axis, define which variables should be plotted).

When you think everything is o.k, you can use the *Preview* function to see, how the chart looks like.

Based on your visual settings, LEONARDO Graph creates a GNUPLOT script (written into the Final Script window) and passes it to the GNUPLOT program. GNUPLOT will then (hopefully) create a low-resolution PNG file based on the script and the data files referred to in the script.

This preview file is then automatically loaded and displayed in the LEONARDO Graph interface.

### **2. Step: Assembling My Script**

If you think the preview is fine and you like to keep it and produce a high-resolution file out of it, you need to transfer it to *My Script*. This is your working place and LEONARDO Graph will not touch it ! You can either save the complete plot sequence to My Script (which will create a script sequence more or less equal to the preview script) or you can add only the plot command to the script (in case you want to plot multiple charts with one file by keeping the other settings)

### **3. Step: Creating the final chart**

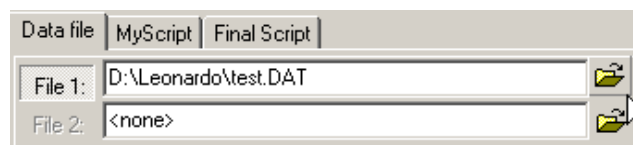
When My Script is completed, you can create the final high-resolution graphic file out of it. You can select the output format for the file and the file name in LEONARDO Graph. After you have pressed the "Make graphic files" button, you have a fresh high-quality graphic file on your disc!

## 2 Step-by-Step example

This section illustrates the typical steps when producing a chart diagram with LEONARDO Graph on a step-by-step basis.

As an example file, we use a 1D inflow profile from ENVI-met. This is a simple 1D text file which plots a number of variables versus the height z. LEONARDO Graph is able to identify this kind of file and automatically uses the correct settings. However, we will not use this function here in order to show all possible steps explicitly.

### 1. Open the file



We select “All other files” as file type, as we do not want to use one of the pre-defined chart styles. LEONARDO opens the file and tries to interpret the file structure.

The file we have chosen looks (in extracts) like that:

```
# Z U V T q Ri Km E Eps ...  
0.20 0.9971 0.9971 293.000 7.09768 0.00 0.02 0.1404 0.1052  
0.60 1.3379 1.3379 293.000 7.09556 0.00 0.05 0.1471 0.0398  
(...)
```

If LEONARDO Graph does not identify some pre-defined file structure, the software assumes that the first line is the *comment line* holding the names for the columns. Each column in the datafile is supposed to be separated with “space” from the other. The comment line is supposed to start with a “#” forcing GNU PLOT to ignore it in the later process.

Please note, that LEONARDO is only interpreting this files. The creation of the final graphs will be done by GNU PLOT and therefore they need to follow the GNU PLOT syntax rules. GNU PLOT can interpret files much more flexible than LEONARDO does at the moment (for example GNU PLOT allows to define other column separator that spaces). Even if the file is displayed totally wrong in LEONARDO, it might still produce correct outputs....

The result of the file parsing process is displayed in the data grid:

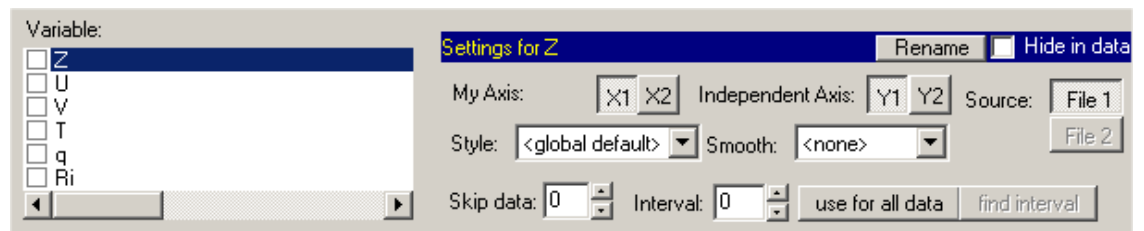
Z	U	V	T	q	Ri	Km
0.20	0.9805	0.9805	292.926	9.93200	0.00	0.02
0.60	1.3165	1.3165	292.911	9.53692	-0.01	0.05
1.00	1.4822	1.4822	292.904	9.34233	-0.01	0.08
1.40	1.5920	1.5920	292.900	9.21351	-0.01	0.11
1.80	1.6782	1.6782	292.896	9.11238	-0.00	0.14
3.00	1.8959	1.8959	292.888	8.85756	-0.04	0.21
5.00	2.0800	2.0800	292.883	8.64306	-0.04	0.38
7.00	2.1987	2.1987	292.880	8.50545	-0.04	0.54
9.00	2.2863	2.2863	292.878	8.40452	-0.03	0.70
11.00	2.3554	2.3554	292.877	8.32514	-0.02	0.86
13.00	2.4126	2.4126	292.877	8.25994	-0.01	1.03
15.00	2.4612	2.4612	292.877	8.20474	0.00	1.19
17.00	2.5035	2.5035	292.877	8.15696	0.02	1.36
19.00	2.5409	2.5409	292.877	8.11491	0.04	1.52
21.00	2.5745	2.5745	292.877	8.07738	0.07	1.69
23.00	2.6049	2.6049	292.878	8.04355	0.09	1.85
25.00	2.6327	2.6327	292.879	8.01276	0.12	2.01

File Type: General Data file

For General Data files (which are all files not being known to LEONARDO) it is assumed that the first column is the independent variable and the other columns are the dependent variables.

## 2. Selecting what to plot and how

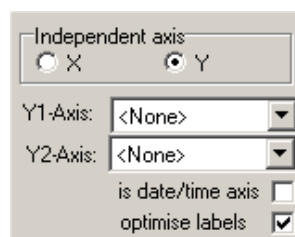
In the lower half of the screen, on the “Data Selection” tab, you can see and select all plot-able columns/ variables:



In the “Variable” list box, you select the variable for which you want to see/change the settings and you chose, if the variable should be plotted in the chart.

In overview, for each variable you can define whether it should be plotted on the X1 or on the X2 axis (if Y is set to be the independent axis) and you can select if Y1 or Y2 should be used as the independent axis. “Style” defines the way the chart series is plotted and “Smooth” sets the level and method of data smoothing.

On the left side of the “Data selection” tab, you find the settings for the axis:

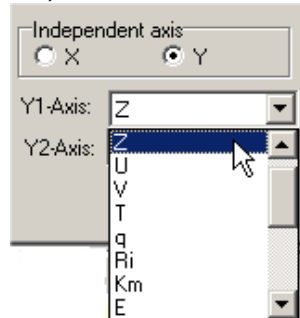


Here you define, whether X or Y is the independent axis and which is the variable (column) used to define the values of the Y1 and (if used) Y2 axis.

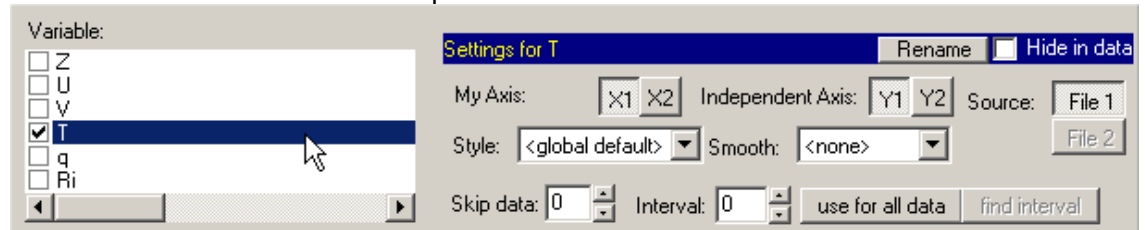
If X is selected to be the independent axis, “X” and “Y” are swapped accordingly.

Let’s now generate a plot which shows the temperature profile (Variable “T” in the file) plotted against the height (variable “Z” in the file). In this case, “Y” is the independent axis and the x-Axis is used to plot the data values. (If we do it the other way round, the chart will be 90° rotated....)

So, we define “Z” as the Y1 axis variable

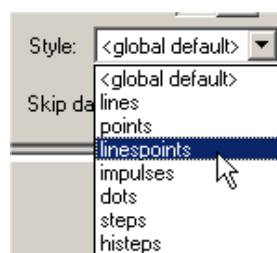


and select the “T” variable to be plotted




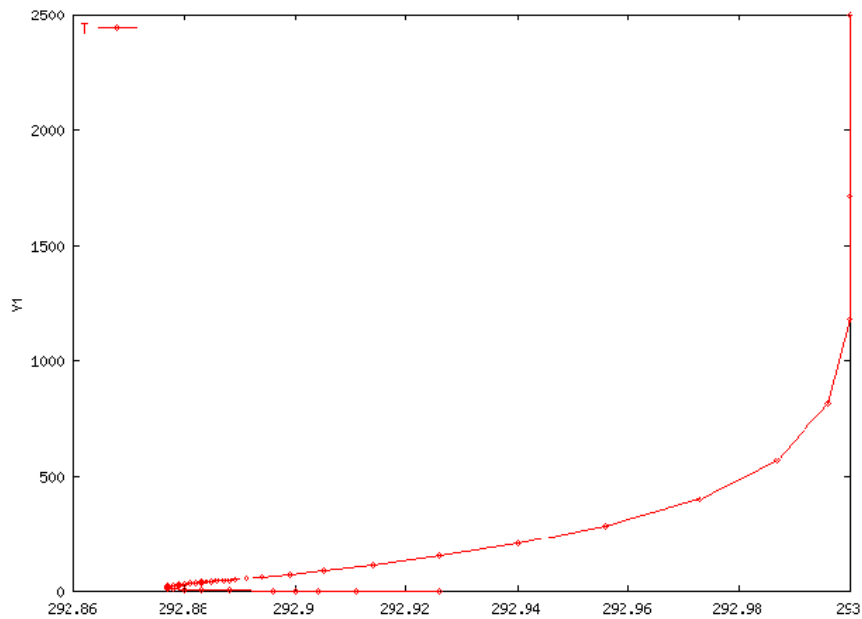
The basic settings are ok: The “T” variable is plotted using the X1 versus the Y1 axis. As we have loaded only one file into LEONARDO Graph at the moment, there is only one option for the source file.

The plotting style “global default” says that this series will be plotted using the default settings for line styles from the “Global Options” tab. We will select the line style “lines with points” for this variable manually here:



### 3. First preview

This is all for the moment. We can now have a look, how the chart will look like by pressing . LEONARDO will now generate a (temporary) script out of your settings, call GNUPLOT and then, after GNUPLOT has generated a low-resolution preview file, load it and preview it (this is the older GIF-based version):



You can control, how the script looks like by selecting the “Final Script” tab:

```

Data file | MyScript | Final Script
set terminal gif
set output "D:/Leonardo/bin/temp_pics/tmp.gif"
cd "D:/Leonardo"
# Enable/disable second x/y axis, Labels -----
set nox2tics
set noy2tics
set xlabel "X1"
set ylabel "Y1"
set x2label ""
set y2label ""
set noxzeroaxis
set noyzeroaxis
set nox2zeroaxis
set noy2zeroaxis
set autoscale x
set autoscale y
set nologscale xyx2y2
# Global linestyle -----
set data style lines
# Settings key (legend) -----
set key left top
# Start plot -----
plot "data.txt" with linespoint

```

This script is automatically generated by LEONARDO Graph and passed to GNU PLOT. It is a fully functional GNU PLOT script and you can copy it and paste it into the GNU PLOT GUI.

If problems happen and GNU PLOT does not produce an output, copying this script into GNU PLOT and start it from there is a good way to find the problem in the script. Unfortunately, the way LEONARDO interacts with GNU PLOT does not provide the option to receive error messages from GNU PLOT. The only thing LEONARDO realizes is, that no output file has been generated, hence an error will have occurred.

#### 4. Fine-tuning

At this point, there are now a million ways how to optimize your diagram. A few ideas are given here:

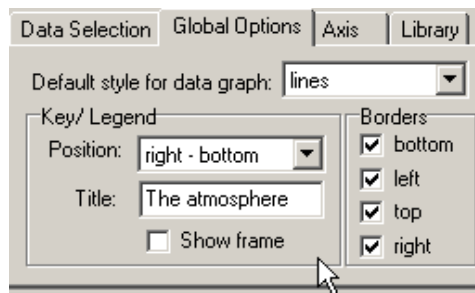
### - Rename the chart series

By default, the names for the series are taken from the first line of the data file. If you don't like e.g. just "T", use the rename button, to rename the column:



### - Enable the key/ legend

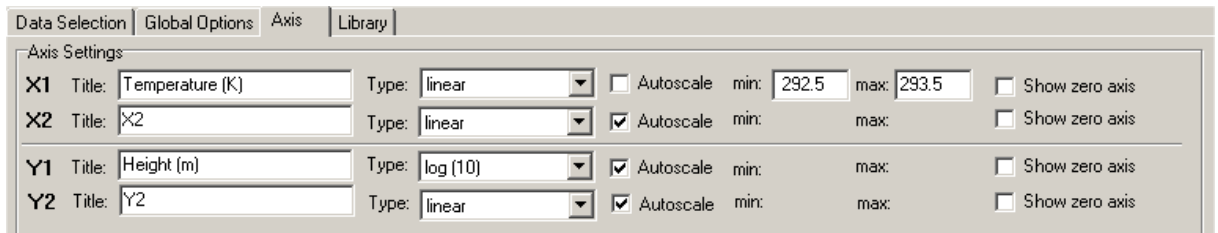
A proper chart provides a key or legend. Go to the "Global Options" tab to organize the legend for the diagram:



Here you can set the title and the position of the chart key and define, if a frame should be plotted.

### - Label and modify axis

Using the "Axis" tab, you can define all required settings for proper axis labeling and properties:



For each of the 4 axis, you can define the "Title" as well as the range and type of axis.

"Autoscale" will fit the axis to the existing data range. Disable this function if you wish to set your own interval. "Show zero axis" will draw a dotted axis through the plot at the position where e.g. X1=0 if you select the X1 zero axis.

## 5. Making the settings permanent in *My Script*

So far, you have only used the "Preview" button to see the effects of changes on the chart. However, in order to make high-quality charts, you need to save your settings by transferring them into *My Script*.

If the chart display seems to be ok, press **Add entire section** to generate the script for this chart in the "My Script" window.

All the information you have entered into the visual interface is now stored as a GNUPLOT script in the "My Script" window:

```

Data file MyScript Final Script
$VER
cd $DATADIR
# Enable/disable second x/y axis, Labels -----
set nox2tics
set noy2tics
set xlabel "Temperature (K)"
set ylabel "Height (m)"
set x2label ""
set y2label ""
set noxzeroaxis
set noyzeroaxis
set nox2zeroaxis
set noy2zeroaxis
set xrange [292.50:293.50]
set autoscale y
set nologscale xx2y2
set logscale y
set logscale y
# Global linestyle -----
set data style lines
# Settings key (legend) -----
set key right bottom title "The atmosphere"
# Start plot -----
plot $DATAFILE using 4:1 title "Temperature " with linespoints

```

This process is in principle the same as using the “Preview” button, but there are some substantial differences:

- the “My Script” script is not a ready-to run GNUPLOT script. It uses LEONARDO internal variables that are not understood by GNUPLOT.
- the “My Script” script will not be erased if you proceed with your work. You can modify it by editing the script inside the window and those changes will be save from any other activities

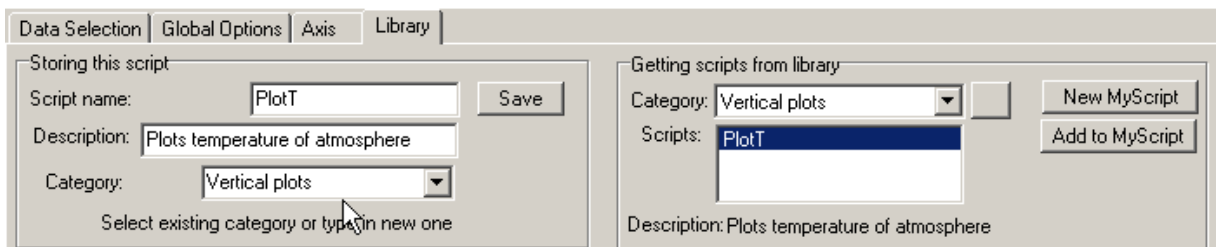
To preview “My Script” you need to press the **Preview MyScript** button rather than the normal “Preview” button.

doing so, LEONARDO will take the “My Script” data, replace the internal variables by absolute references and save it in “Final Script”. Then the same process as before takes place.

## 6. Managing scripts

There are two ways how to save your work in “My Script” for later use:

- a) you can copy & paste it into some larger text file and save this file or,
- b) you can use the “Library” functions supplied by LEONARDO Graph.



The “Library” tab supplies both actions required for script managing: saving scripts into the library and getting scripts out of the library.

All references in the scripts are relative. For example, if you have plotted a diagram based on the data file “ABC.DAT” and save the script, the filename is not stored.

If you open the script the next time, but this time you have a data file called “123.DAT” open, the script will be run for this file.

### Saving the script to the library

First, you have to give your script a name .

All scripts are saved as normal text files with the extension “.esc” in the subfolder \scriptlib of the LEONARDO application directory.

To identify the script, you can enter a **description** saying what this script does.

In order to sort different sorts of scripts, you can assign a **category**, to which this script belongs. You can either select a category out of existing categories or type in a new category. (The list of existing categories is compiled based on the scripts found in the \scriptlib folder and the categories they use)

### Getting a script from the library

To get a script from the library, you select a matching category and then select the script you want to have.

Then, you have two options:

New MyScript

will **delete** (!!!) the “My Script” window and insert the selected script as a new script.

Add to MyScript

leaves your actual “My Script” untouched and will append the selected script to it.

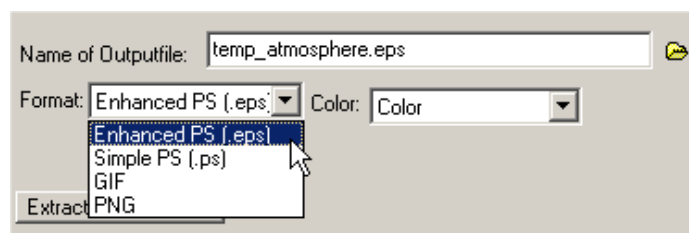
## 7. Creating high-quality outputs

One of the main aims of LEONARDO is to provide high-quality graphic outputs that can be used for publications.

In order to create a high-quality output of your actual chart, you have to do two things:

First, make sure that “My Script” holds the script for the charts as you like it to be (pressing “Preview My Script” will produce a preview version of what you will get).

Then, go to the “Global Options” tab and select the filename and the file format for your output file.

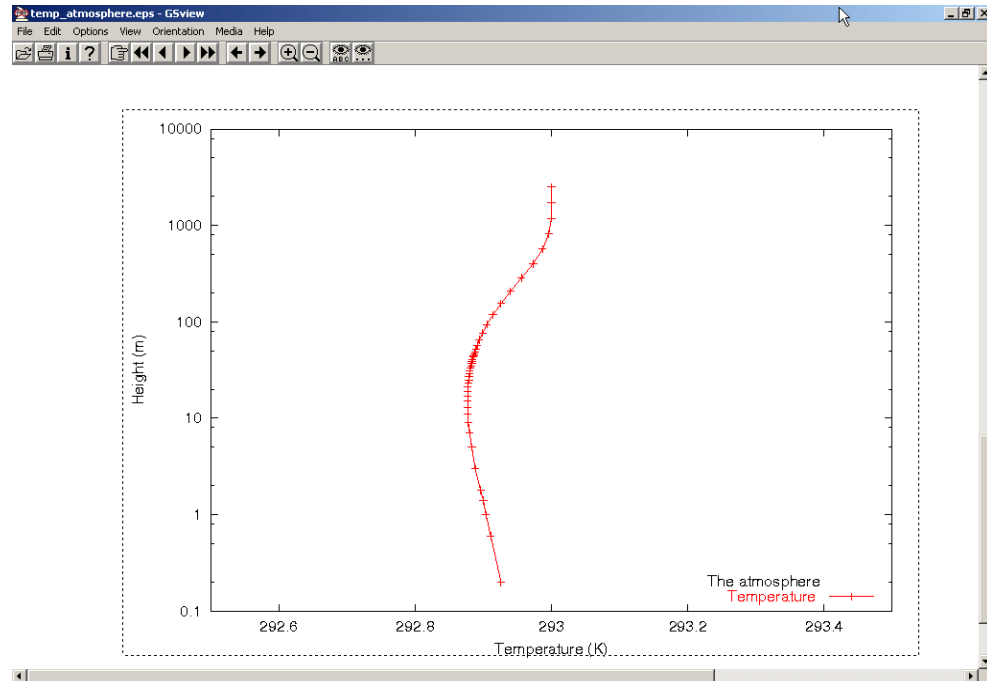


The file formats available in LEONARDO are the basic file formats provided by GNUPLOT. They should serve almost all needs. By default, for traditional publications the Enhanced PS format (eps) should be used. For web sites, GIF and PNG are also fine.

You can select if the output file should be in “Color” or “Monochrome”, for the GIF format, “Grayscale” is also supported.

Finally, press **Make graphics file** to generate the output file..

The output file is not displayed in LEONARDO Graph. Depending on the format, you need additional programs to display the file. For example, GhostScript is recommended for previewing eps and ps files:



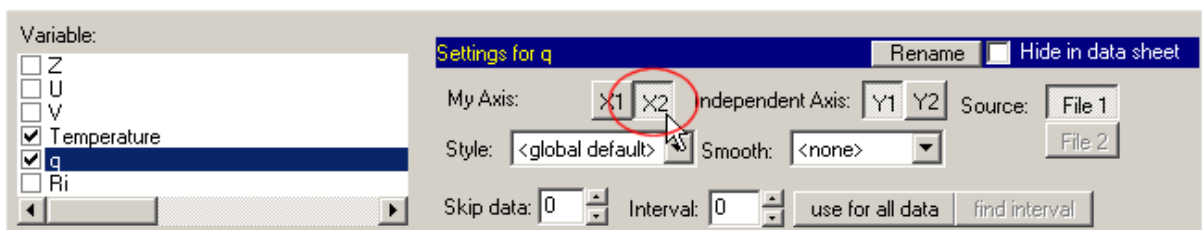
(Please that the point symbols between the PNG preview and the final Postscript file are not the same. GhostScript comes along with an example file showing the different Postscript symbols and their associated index)

## 8. Working with two axis

If more than one variable should be displayed, it can be useful to use more than one dependent or even more than one independent axis for proper display.

For example, if we wish to display the humidity of the atmosphere (variable “q”) at the same time with the temperature, we need to use the second X axis (X2) for a proper display.

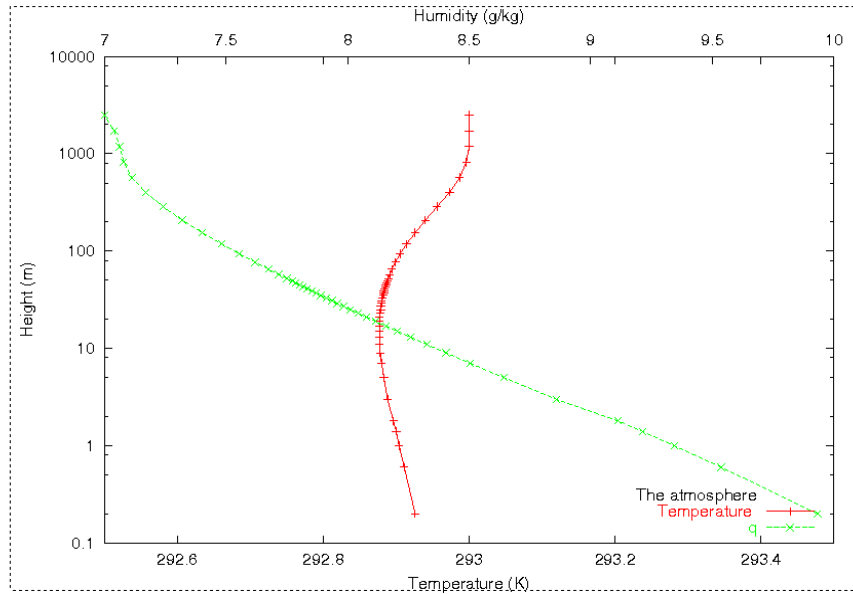
To select the axis, go to the “Data Selection” tab and click on the variable “q”:



In the settings section, select now “X2” as “My Axis” for q.

Then repeat the steps already introduced to label and scale the axis and so on.

The results might then look like this (eps-Version):



The same concept applies if a second independent axis (Y2 in this case) should be used. You can disable axis by using the "Borders" option on the "Global Options" tab.

## 9. Working with two files

The second feature which is often required is to compare the state of a variable in different files. LEONARDO directly supports the usage of two data files in the visual interface. However, this function only works if these two files have exactly the same format. If your variable is in column 3 in data file 1 and in column 5 in data file 2, it will not work (in this case, you need to edit *My Script* manually in order to point GNUPLLOT towards the correct column (see also Step 10 in this manual).

Let's continue with the example above, but add a second data file to the chart:

Data file | MyScript | Final Script

File 1: D:\Leonardo\test.dat

File 2: D:\Leonardo\test2.DAT

Z	U	V	T	q	Ri	Km
0.20	0.9971	0.9971	293.000	7.09768	0.00	0.02
0.60	1.3379	1.3379	293.000	7.09556	0.00	0.05
1.00	1.5061	1.5061	293.000	7.09451	0.00	0.08
1.40	1.6177	1.6177	293.000	7.09381	0.00	0.12
1.80	1.7054	1.7054	293.000	7.09326	0.00	0.14
3.00	1.9268	1.9268	293.000	7.09188	0.00	0.22
5.00	2.1143	2.1143	293.000	7.09071	0.00	0.38
7.00	2.2353	2.2353	293.000	7.08995	0.00	0.54
9.00	2.3245	2.3245	293.000	7.08939	0.00	0.71
11.00	2.3951	2.3951	293.000	7.08894	0.00	0.87
13.00	2.4535	2.4535	293.000	7.08857	0.00	1.04
15.00	2.5031	2.5031	293.000	7.08826	0.00	1.21
17.00	2.5463	2.5463	293.000	7.08799	0.00	1.37
19.00	2.5846	2.5846	293.000	7.08774	0.00	1.54
21.00	2.6189	2.6189	293.000	7.08752	0.00	1.70
23.00	2.6499	2.6499	293.000	7.08732	0.00	1.87
25.00	2.6784	2.6784	293.000	7.08714	0.00	2.04
27.00	2.7045	2.7045	293.000	7.08697	0.00	2.20
29.00	2.7288	2.7288	293.000	7.08682	0.00	2.36
31.00	2.7514	2.7514	293.000	7.08667	0.00	2.53
33.00	2.7726	2.7726	293.000	7.08653	0.00	2.69

File Type: General Data file

You can switch in the display between both data files by pressing the buttons labelled “File 1” and “File 2”.

For both variables, “T” and “q” you now have to select “File 2” in the properties selection:

Variable:

- Z
- U
- V
- Temperature
- Humidity
- Ri

Settings for Temperature Rename  Hide in data sheet

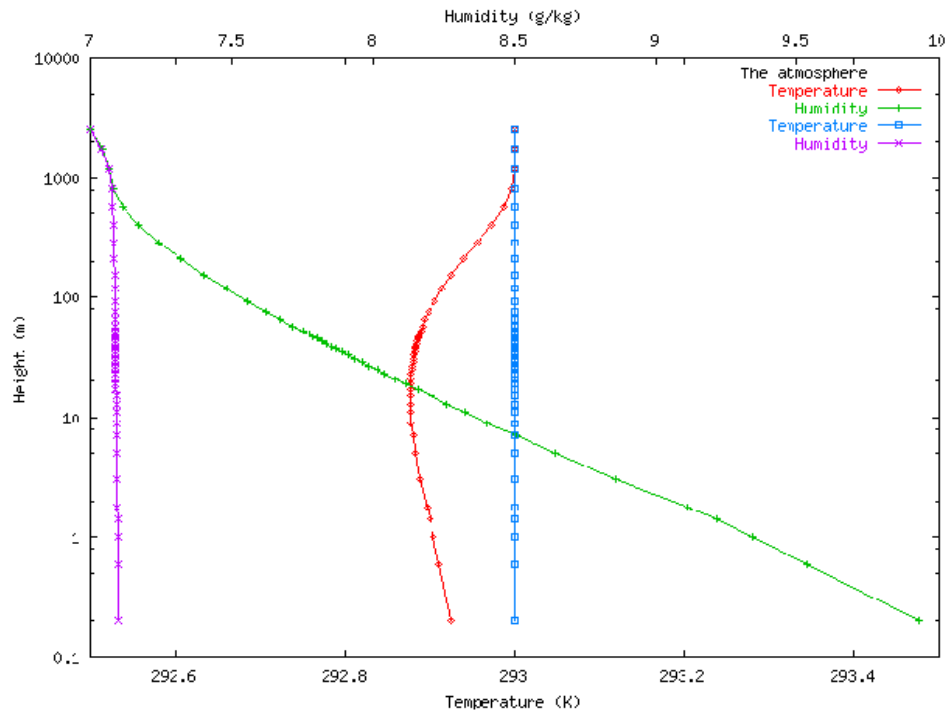
My Axis:   Independent Axis:   Source:

Style:  Smooth:

Skip data:  Interval:

Please note, that all other settings, including the label, will now apply to both plots, for File 1 and for File 2. The recent version of LEONARDO Graph does not support individual labels for variables coming from different files. If you wish to use different labels or styles (which would make sense) you need to edit “My Script”!

After previewing and adjusting the things that need to be adjusted we get the following preview:



which is based on this “My Script”:

```

Data file  MyScript  Final Script
$VER
cd $DATADIR
# Enable/disable second x/y axis, Labels -----
set x2tics
set noy2tics
set xlabel "Temperature (K)"
set ylabel "Height (m)"
set x2label "Humidity (g/kg)"
set y2label ""
set noxzeroaxis
set noyzeroaxis
set nox2zeroaxis
set noy2zeroaxis
set xrange [292.50:293.50]
set autoscale y
set autoscale x2
set nologscale xx2y2
set logscale y
set logscale y
# Global linestyle -----
set data style lines
# Settings key (legend) -----
set key right top title "The atmosphere"
# Start plot -----
plot $DATAFILE using 4:1 title "Temperature" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity" with linespoints,\
    $DATAFILE2 using 4:1 title "Temperature" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity" with linespoints

```

## 10. Fine-tuning the script

Continuing from the example before, we have seen that LEONARDO Graph does not support different titles for variables coming from different files. However, looking at the “My Script” script generated by LEONARDO, it is obvious, that this problem can be solved quickly, even if we do not know anything about the GNUPLOT scripting language so far.

The final plot command reads:

```

plot $DATAFILE using 4:1 title "Temperature" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity" with linespoints,\
    $DATAFILE2 using 4:1 title "Temperature" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity" with linespoints

```

Obviously, the first sequence after \$DATAFILE defines the labels and styles of the variables coming from the first data file and the sequence after \$DATAFILE2 does the same thing for the second data file. The title for the series is the same for both files, and this needs to be changed.

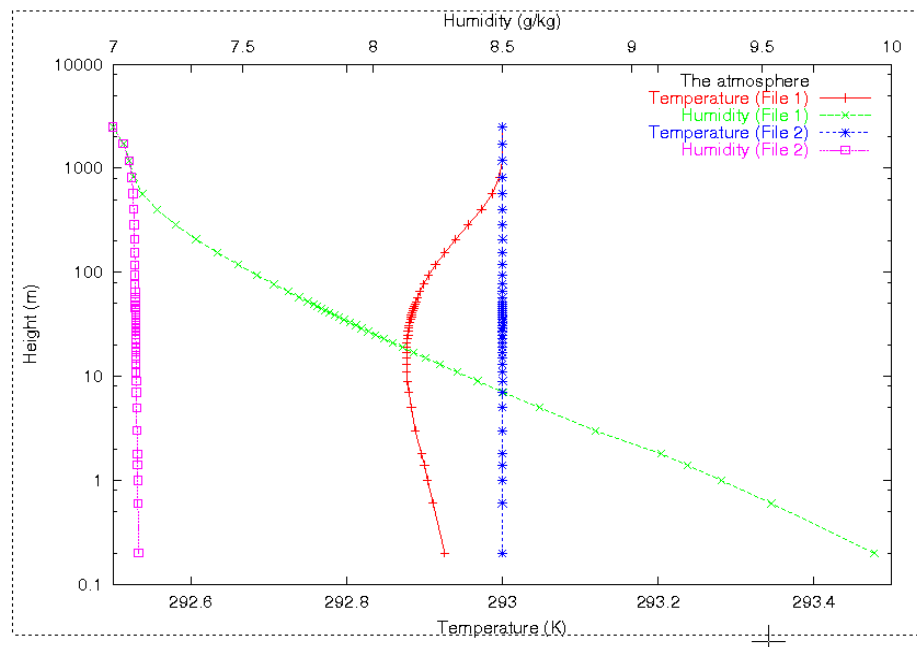
So we just go into the “My Script” window and **modify** what is written there (remember that everything you write there is save from any actions, except you delete it by yourself):

```

plot $DATAFILE using 4:1 title "Temperature (file 1)" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity (file 1)" with linespoints,\
    $DATAFILE2 using 4:1 title "Temperature (file 2)" with linespoints,\
    "" using 5:1 axes x2y1 title "Humidity (file 2)" with linespoints

```

Then, after pressing “Preview My Script” or generating a graphic file, we get (as eps-version):



When you work with LEONARDO Graph and GNUPLOT, you will discover, that the click-and-point interfaces are not always the easiest solution. Sometimes it is much faster and powerful to use some script commands!

Like we have changed the title of the variable series here, you can do the same with any other command that is supported by GNUPLOT, but not featured as a visual element in LEONARDO. For example, you can plot variables from more than 2 files in one diagram by just adding some more lines to the plot command (but you must name the files explicitly as LEONARDO does only support the variables \$DATAFILE1 and 2). Also, you can fix problems if the data structures of the files are not the same, e.g. the same variable is in different columns).

However, modifying the script directly requires (except some obvious things like the example before) to consult the GNUPLOT manual. GNUPLOT is a very strong charting engine and LEONARDO Graph does only help in generating the bottom functionality for a GNUPLOT script.

