Manual for wxChart 1.0: a Chart layout library for wxWindows

Neil Dudman

August 1995

Contents

1. Introduction

This manual describes a chart drawing class library for wxWindows. It provides the start of a graphing tool; currently wxChartLayout supports the layout of bar, line, area, curve floating bar, percent bar, pie, xyplot. There are a number of user definable options to minor and major tickmarks, 3D display (bars only) etc. Multiple datasets may be defined per chart, each of which may have different display types.

wxChartLayout is an abstract class that must be subclassed. The programmer defines various member functions which will access whatever data structures are appropriate for the application, and wxChartLayout uses these when laying out the chart.

wxStoredBarLineChart is a class derived from wxChartLayout that may be used directly to draw data in a chart on a canvas. It supplies storage for the data in a two dimensional array and draws to a device context of the Canvas passed to the constructor function.

Below are the example charts generated by the program test.cc.

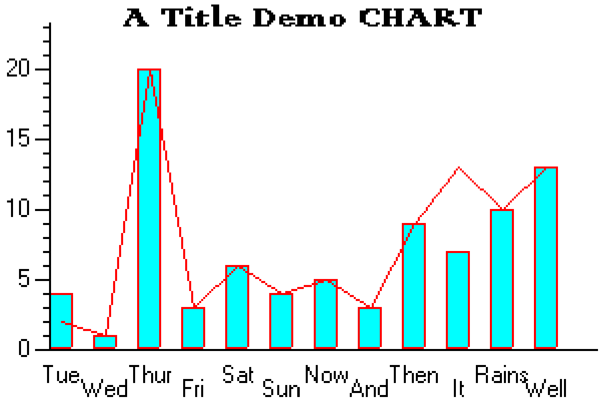

Figure 1: Bar and line charts

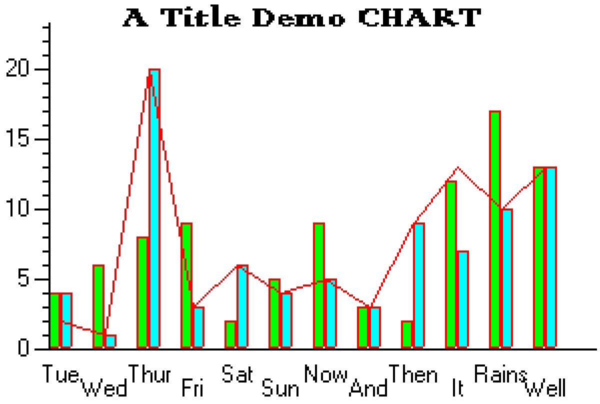

Figure 2: Bar2 and line charts

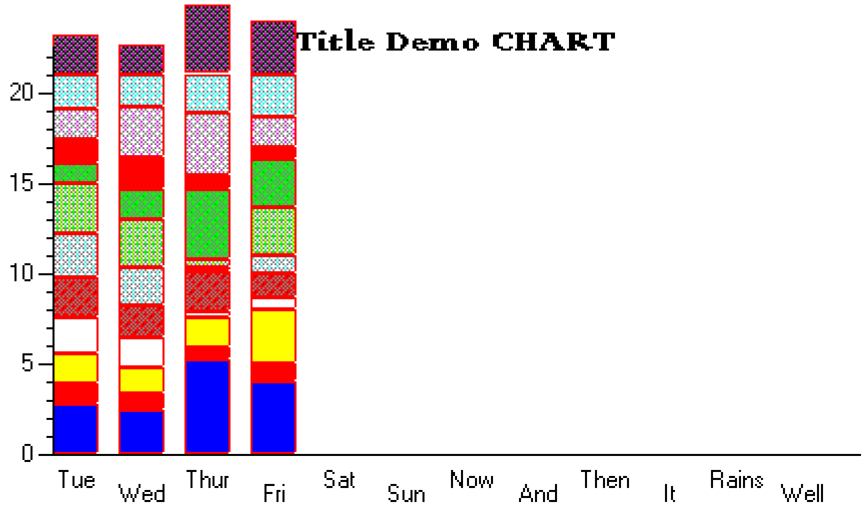

Figure 3: Floating bar chart

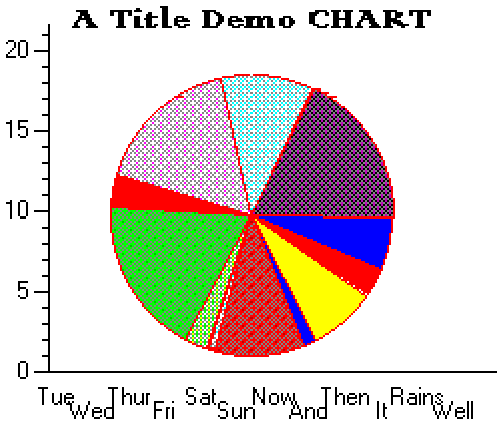

Figure 4: Pie chart

1.1. Things I would like to add

- Add Legend for Pie, Percent Bar charts
- Use a spread Sheet widget

2. The Test Program

The test program included with the chart library uses the shapes code written by Julian Smart as part of wxBuilder and allows the chart to be resized and moved by selecting the object and dragging the corners. The right mouse button may also be used to bring up a simple frame where various settings of the graph may be changed (not yet complete).

All the types of charts that the libarary supports have been added to the test program as seperate datasets. It is posible to turn off various datasets by checking various menu options.

3. Using wxChartLayout

Include the file `lbchart.h` in your source, and link with the libary `wxchart.lib` (Windows) or `libwxchart.a` (UNIX).

4. Class reference

The member functions are given in alphabetical order except for the constructors and destructors which appear first.

4.1. wxChartLayout: wxObject

This abstract class is used for drawing a Chart. You must derive a new class from this, and define member functions to access the data that wxChartLayout needs. See the functions *wxStoredChart::GetData* and *wxStoredChart::SetData* in a derived class that I have written for the test program included with this library: wxStoredBarLineChart.

The application should call *Draw* to Draw the Chart.

wxChartLayout::wxChartLayout

void wxChartLayout(**wxCavas ****canvas = NULL*)

Constructor.

wxChartLayout::AddDataSet

int AddDataSet(**int** *dataType*, **int** *row*)

int AddDataSet(**int** *dataType*, **wxList** *rows=-1*)

Used to Add a new dataset to a Chart.

dataType is one of

- wxBar
- wxLine
- wxArea
- wxCurve
- wxFloatingBar
- wxXYPlot
- wxPie
- wxPercentBar

Use **row** for single row **dataType** and use **rows** for a list of rows when using multiple row data types like wxPercentBar, wxFloatingBar.

wxChartLayout::Draw

void Draw(**void**)

Call this to let wxChartLayout draw the chart itself. The device context must have been set in the constructor or using SetDC.

wxChartLayout::Get3D

Bool Get3D(**void**)

wxChartLayout::GetOrientation

int GetOrientation(**void**)

Returns the current orientation of the chart, wxHORIZONTAL or wxVERTICAL.

wxChartLayout::GetDataStyle

wxBrush * GetDataStyle(**void**)

wxChartLayout::GetData

float GetData(**int** *col*, **int** *row*)

Virtual function to allow access to data using **col** and **row**. This must be defined in your class derived from wxChartLayout.

wxChartLayout::GetLabel

char * GetLabel(**int** *col*, **int** *row*)

Virtual function to allow access to data **Label** using **col** and **row**. This must be defined in your class derived from wxChartLayout.

wxChartLayout::SetEdgeTop

void SetEdgeTop(**int** *top*)

Sets the maximmum edge from the **top** that the data in graph may be drawn to.

wxChartLayout::SetEdgeBottom

void SetEdgeBottom(**int** *bottom*)

Sets the maximmum edge from the **bottom** that the data in graph may be drawn to.

wxChartLayout::SetEdgeLeft

void SetEdgeLeft(**int** *left*)

Sets the maximmum edge from the **left** that the data in graph may be drawn to.

wxChartLayout::SetEdgeRight

void SetEdgeRight(**int** *right*)

Sets the maximmum edge from the **right** that the data in graph may be drawn to.

wxChartLayout::SetTitle

void SetTitle(**char ***title*)

Allows the title of the chart to be set.

wxChartLayout::GetTitle

char * GetTitle(**void**)

Returns the title of the chart.

wxChartLayout::SetTitleFont

void SetTitleFont(**wxFont** **titlefont*)

Allows the title font to be set.

wxChartLayout::GetTitleFont

wxFont * GetTitleFont(**void**)

wxChartLayout::SetMaxWidthHeight

long SetMaxWidthHeight(**int** *width*, **int** *height*) Sets the maximmum **width, height** of the chart. This area includes the edges which are not drawn in by the data and reserved for labels, titles, etc.

wxChartLayout::SetChartCenter

void SetChartCenter(**int** *x*, **int** *y*) Sets the center coordinates of the chart.

wxChartLayout::SetStartEndCol

void SetStartEndCol(**int** *colStart*, **int** *colEnd*)

This should be a virtual function, but it basically sets the range of columns that the chart gets the data to display from. For example 8-20 will plot data from column 8 to column 20.

wxChartLayout::SetDataSetOrder

This is not currently implemented but I intend to support prioritizing the layout of datasets so that the layout may be specified (somehow).

wxChartLayout::SetMajorTickInc

void SetMajorTickInc(**int** *inc*)

Sets the increment size for **Major** tick marks.

wxChartLayout::SetMinorTickInc

void SetMinorTickInc(**int** *inc*)

Sets the increment size for **Minor** tick marks.

wxChartLayout::GetMajorTickInc

void GetMajorTickInc(**void**)

Gets the increment size for **Major** tick marks.

wxChartLayout::GetMinorTickInc

int GetMinorTickInc(**void**)

Gets the increment size for **Minor** tick marks.

wxChartLayout::SetTailText

void SetTailText(**char ****text*)

Allows an optional text string to appended to the data values. Could be used to put a percentage sign after each value on the axis.

wxChartLayout::SetHeadText

void SetHeadText(**char** **text*)

Allows an optional text string to prepend to the data values. Can't think of any use but you may find it useful.

Gets the increment size for **Minor** tick marks.

wxChartLayout::GetTailText

char * GetTailText(**void**)

Returns the text tail text of value labels the axis.

wxChartLayout::GetHeadText

void GetHeadText(**void**)

Returns the text header text of value labels

Gets the increment size for **Minor** tick marks.

wxChartLayout::SetTickStyle

void SetTickStyle(**int** *style*) Valid styles are:

- wxTickIn
- wxTickOut
- wxTickNone

wxChartLayout::GetTickStyle

int GetTickStyle(**void**) Returns the tick style of the axis. See *SetTickStyle* for valid **styles**.

wxChartLayout::ShowValues

float ShowValues(**Bool** *bool*) Determines whether the data values of bars are displayed inside the bar.

wxChartLayout::GetShowValues

Bool GetShowValues(**void**) Returns TRUE if values are currently displayed inside bars.

wxChartLayout::SetOrientation

void SetOrientation(**int** *orient*) Valid chart orientations are wxHORIZONTAL and wxVERTICAL.

wxChartLayout::SetDataType

void SetDataType(**int** *dataset*, **int** *datatype*)

Sets the graph type of the supplied dataset. See *wxChartLayout::wxChartLayout* (page 5).

wxChartLayout::SetLineStyle

void SetLineStyle(**int** *dataset*, **wxPen ****pen*)

Sets the line style (pen).

wxChartLayout::GetLineStyle

wxPen * GetLineStyle(**void**)

Returns the current pen.

wxChartLayout::SetDataStyle

void SetDataStyle(**int** *dataset*, **wxBrush** ***brush**)

Sets the brush used for drawing the data.

wxChartLayout::Set3D

void Set3D(**Bool** *bool*)

Sets 3D display mode.

wxChartLayout::SetData

void SetData(int *col*, **int** *row*, **float** *value*) Virtual function to allow access to data using **col** and **row**. This must be defined in your class derived from wxChartLayout.

wxChartLayout::SetLabel

void SetLabel(**int** *col*, **int** *row*, **char** ***label**)

Virtual function to allow access to data **Label** using **col** and **row**. This must be defined in your class derived from wxChartLayout.

Index