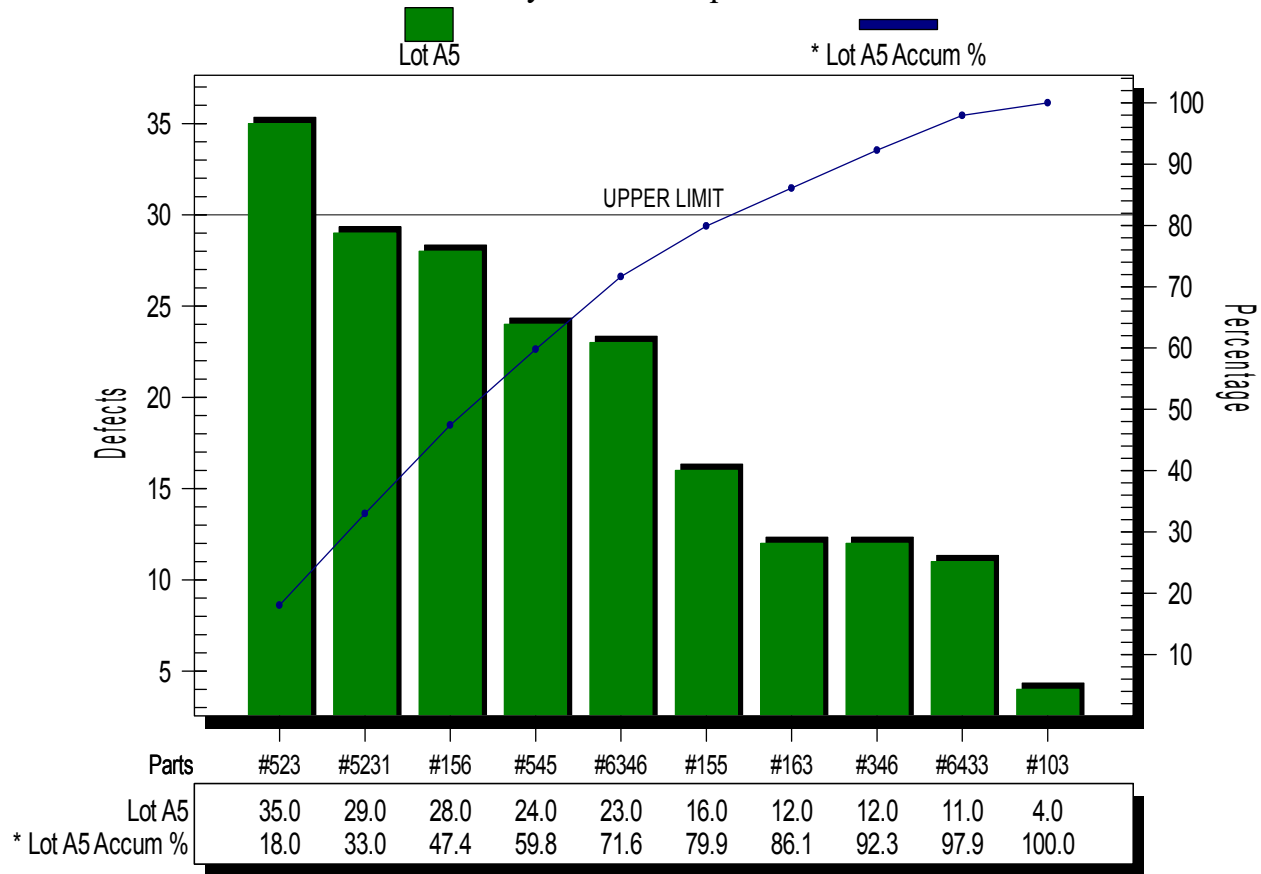


# GigaSoft™ ProEssentials™ v1.5

Windows 3.1 DLL, VBX, and FLL for Presentation Charting

## Weekly Defects Report



***If your serious about producing quality software. Then you seriously need ProEssentials for your graphing needs. No memory/resource leaks, no mysterious crashes, no wasted time, guaranteed. What else?...***

- \* Images prepared in memory so no flashing redraws, even after you make programmatic changes.***
- \* Intelligent image construction. This means no overlapping text independent of shape, size, style, or amount of data. You or your users won't have to tweak images to make them look right. ProEssentials is the only tool in the world that has this important feature. It's especially important if you let your users change the graph's shape or style.***
- \* Integrated Data Grid that horizontally pans with the graph.***
- \* Extensive export capabilities which includes the spawning of OLE objects.***
- \* Comprehensive and Automatic Hot-Spot / Drill-Down mechanism.***
- \* Integrated Zooming capabilities.***
- \* Smooth Real-Time that will run indefinitely while others crash within hours.***
- \* Null-Data handling.***
- \* Integrated Maximization capabilities.***
- \* Revolving Subset and Panning Point functionality with smooth image movement.***
- \* User Customization Dialogs and Floating Popup Menus.***
- \* Support for VB, VC++, BC++, Delphi, Clarion, Gupta, FoxPro, PowerBuilder and others.***
- \* Absolutely the easiest graphing tool to learn, implement, and maintain.***

VB Tech Journal says... **“I found ProEssentials both professional and essential. It’s a first-class product.”**

## **Table of Contents**

1. Introduction
2. Functionality Summary
3. Code Examples
4. Conclusion

### **1. Introduction**

The ProEssentials are three controls designed to provide graphic presentation functionality quickly and easily, yet the end-results look like it took you months. All that is needed is to give these controls data and your done. If you want to alter how the ProEssentials look or operate, you adjust properties, no complex functions are needed. The ProEssentials are a stable, professionally engineered tool that will help your projects be visually and functionally impressive while being delivered on time. The cost of a development tool is not truly realized until the tool is implemented. We feel confident that the ProEssentials represents one of the best tool values in the industry.

The ProEssentials consist of:

**GRAPH:** line, bar, point, area, area stacked, area stacked%, bar stacked, bar stacked%, points plus line, points plus best fit line, points plus best fit curve (least squares of varying degrees), points plus spline, spline, histogram, high/low bar, high/low line, high/low/close, open/high/low/close, and box plot. Designed to show quantitative /categorized real variables with respect to the Y Axis.

**SCIENTIFIC GRAPH:** line, point, stick, points plus line, points plus best fit line, points plus best fit curve (least squares of varying degrees), points plus spline, spline, and bubble. Designed to show plots of real variables vs other real variables with respect to both the X and Y Axes.

**PIE CHART:** Standard pie chart designed to show percentages.

## 2. Functionality Summary

Rather than cramming all types of functionality into one control, the ProEssentials break out graphic functionality into three controls. This makes the controls more logical and user friendly by only offering customizations unique to a graphing situation. All functionality resides in one DLL with three small VBX alternate interfaces. When implementing the VBX interfaces, you can still call DLL calls to bypass the VBX interface. This lets you get the best of both worlds, VBX for visual integration and default property initialization, DLL calls to pass large amounts of data instantly.

### 1) The ProEssentials have the **BEST** built in user interface:

- The ProEssentials have their own customization dialogs, colors dialog, fonts dialog, print dialog, image export dialog, and text export dialog.
- **Double-Clicking** the ProEssentials, the user is shown a customization dialog that allows the object to be customized, maximized, and exported.
- **Right-Button-Clicking** the ProEssentials, the user is shown a floating popup menu allowing a quick alternate interface to much of the object's customizations and functionality.
- Pressing keystrokes, the ProEssentials provide the fastest **short-cuts** to the object's customizations and functionality.
- The ProEssentials function like a button, they have a **focal-rect** and can be tabbed to and from. The user interface can be adjusted or bypassed. The ProEssentials can also be used to provide **background image construction**.
- The ProEssentials have the best **hot-spot/drill-down mechanism**. Possible Hot-Spots include, subset labels, point labels, data points, graph coordinate feedback, and table coordinate feedback. When the cursor moves on top of a Hot-Spot, it changes to the same Hot-Spot cursor used in Windows Help. The developer can respond to both click and double-click hot spot events. The Graph and Scientific Graph Objects also have a built in mechanism to prompt the user of data point values.

### 2) The ProEssentials show the **MOST** attention to detail:

- The Graph Object and Scientific Graph Object both have a **scrolling / comparison subset mechanism**. An object can contain multiple subsets worth of information, yet (1) the user can choose what/how much of the information is shown at any one time, and (2) the user can revolve through subset information while comparing individual subsets to other permanent subsets.
- The ProEssentials work with three types of subset information: Normal Subsets, Comparison Subsets, and Right Axis Comparison Subsets. **Comparison subsets** are special subsets which are normally graphed as a thin line. You can combine a line/bar/area/...etc graph with a comparison line graph. Comparison subsets also have the option to be graphed as lines/bars/areas/...etc. **Right Axis Comparison Subsets** are the same as Comparison Subsets, however, they are plotted with respect to an alternate Y Axis scale which is on the right side of the graph.
- The Graph Object has **scrolling / random point capabilities**. The developer and user both have control over the amount of point information that is displayed at any one time. If the amount of point information displayed is less than the amount contained in the object, there will be a horizontal scrollbar allowing the user to view the rest of the data. When a Graph Object displays a large quantity of point information, the developer can define **alternate meaningful frequencies** to be used to place point labels and tabled information. For example, if you have 1500 minutes worth of data, you can define alternate frequencies of 5minutes, 15minutes, 1/2 hour, and 1hour. So no matter how many points are being

displayed, there will be point labels and tabled data at an optimum frequency to produce legible text.

- The Graph Object supports a Y and Right Y Axis Log Scale and the Scientific Graph Object supports a **X, Y, and Right Y Axis Log Scale**. All plotting methods function with log scaling, even the best fit lines and curves.
- The Graph and Scientific Graph Objects have powerful **zooming capabilities**. The mouse is used to control a rubber-band selection rectangle which designates the new extents of the graph. The developer can control whether zooming capabilities are horizontal, vertical or both horizontal and vertical. The Graph Object implements horizontal zooming capabilities as a short cut method of selecting and initializing scrolling points parameters. All Hot Spots and Coordinate Prompting remain in effect while zooming. The Scientific Graph allows programmatic control of zooming functionality.
- The ProEssentials **support NULL data**. This feature allows for missing data and multiple subsets with varying amounts and frequencies of data.
- The Graph Object has **real-time capabilities**. The Graph Object has the ability to automate the shifting/ appending of data, and scrolling the new information into view. This features is ideal when your gathering timely data and need to display the most current image. When the Graph Object receives new data, it understands to regenerate and invalidate only the graph portion of the image. All object customizations by either dialog or popup menu remain functional during real-time operation. Combined with the fact that images are prepared in memory, all these features allow for real-time implementations which are very professional.
- The Graph Object has the capability of automatically producing a large variety of **statistical comparison subsets**. The Graph Object will **automate the construction of Pareto style** charts by sorting the data and generating the accumulated percent line as a comparison subset.
- The Graph and Scientific Graph Objects have the **best automatic grid tick / line frequency selection**. There is also **manual control over grid tick / line frequencies**. V1.5 has **improved the automatic grid construction** to better handle data values in the range of .001 to 1.
- The Graph Object can display a graph, **graph plus table**, or just the table. The table can include only those subsets currently graphed, or show all subset information contained in the object.
- The Pie Chart Object is ideal for showing unpredictable data because it will **group small sliver type slices** into an \*Other slice. A footnote is placed at the bottom of the control notifying the user what the \*Other slice contains.
- For Point style graphs, the Graph and Scientific Graph Objects allow the selection of **12 point styles in three variations of size**. These same point styles are also used as symbols when a MonoChrome plus Symbols image is generated.
- The Graph and Scientific Graph Objects allow the selection of **7 line styles**.
- The Graph and Scientific Graph Objects have **upper/lower control limits** which can be superimposed onto the graph's grid.
- The ProEssentials provide both a **default and a custom set of properties**. The user can make customizations and then toggle back and forth between the original and custom parameter sets. The developer can also initialize the custom parameter set so that the user only needs to press a key to transform the image into an alternate common form.
- The ProEssentials have built in **maximization capabilities**. A small control on a form can be quickly maximized by the user.
- The ProEssentials have built in **context sensitive help**. There are API calls that the

developer can call to receive context IDs that correspond to ProEssentials based windows. This allows for an integrated help system. The ProEssentials even come with everything necessary to adjust the help file text to target a particular audience.

- The ProEssentials print dialog allows the printing of **multiple pages**. Varying amounts of subset information can be automatically placed onto separate pages.
- During development, the ProEssentials will transmit debug messages to both DBWIN and MS Visual Basic's immediate window. You don't have to look at function return values (and then look the return values up in a book) to know if your making a mistake. **The ProEssentials will tell you.**
- The ProEssentials provide **40 API calls** which form a logical and easy to comprehend programming interface. There are so many properties I don't want to count them. SDK developers will appreciate that they have serialization capabilities.
- The ProEssentials come with a 150 page manual. The manual's text and reference sections are written in a **clear and familiar format**. SDK Property names contain data-type nomenclature.

### 3) The ProEssentials have the **SMARTEST** image construction:

- The ProEssentials **prepare their images in memory**. Scrolling functionality is professionally fast and clean. Your users wont get a headache watching their screen repeatedly redraw. You can also make programmatic changes to the image and your users will not see any redrawing.
- The ProEssentials provide a method to **quickly transform an image from color, to monochrome, to monochrome with symbols**. You can instantly get an image intelligently designed for printing and your users will have no problem distinguishing subsets or other graph attributes.
- The ProEssentials **scale all image attributes based on one property, FontSize**. Small controls placed onto a form can default to a relatively large font. Images exported to the printer can have a relatively small font. This feature instantly produces high quality images under varying situations.
- The ProEssentials allow the user to **adjust the precision** of tabled data, data labels, and coordinate prompting. If the object is a small control on a form, the precision can be set to zero which will cause the text to be larger and more readable. The user can always maximize the object and increase the precision when necessary.
- The ProEssentials **check for overlapping text**. Subset Labels, Point Labels, and Tabled Data are all sized to fit and Data Point Labels are relocated to prevent overlapping text. Overlapping labels are embarrassing and frustrating. The ProEssentials will not embarrass you.
- The Scientific Graph Object can place **several types of labels next to data points**. (1) automatic data point value labels, (2) point labels, and (3) individual data labels for when more than one subset is displayed. The ProEssentials will attempt to not allow these labels to overlap.
- If Graph Object point labels get too small, the point labels are **automatically forced into vertical orientation**. The user can also adjust point label orientation.

#### 4) The ProEssentials have the **MOST** export capabilities:

- Export a **metafile** to the clipboard, file, or printer (via a print dialog.)
- Export a device independent **bitmap** to the clipboard or file.
- Export varying amounts of **text data** in various formats (via a text export dialog) to the clipboard or file.
- Export an **OLE object** to the clipboard. The object is served by an OLE-miniserver which is also redistributable. Imagine the power and convenience of allowing an exported image to retain all the functionality listed on this sheet.

### 3. Code Examples

The ProEssentials are extremely easy to implement. Our current customers have had no problem in quickly getting data into the ProEssentials.

#### SDK Example

```
float  fData[4][20];                /* 4 subsets by 20 points */
char   szSubsetlabels[] = "Texas\tCalifornia\tFlorida\tWashington\t";
int    i, j;

/* hWndPE is a global variable to hold handle to Object */
hWndPE = PEcreate (PECONTROL_GRAPH, WS_VISIBLE, &rect, hWnd, 10);
if (!hWndPE) { /* if PEcreate returns Zero something went wrong */ }

PEnsset (hWndPE, PEP_nSUBSETS, 4);      /* Object will manage 4 subsets */
PEnsset (hWndPE, PEP_nPOINTS, 20);     /* Object will manage 20 points */
PEnsset (hWndPE, PEP_bPREPAREIMAGES, 1); /* Prepare images in memory */
PEnsset (hWndPE, PEP_nDATAPRECISION, 1); /* Table Data has 1 decimal*/

for (i=0; i<4; i++)                  /* make some random data */
    for (j=0; j<20; j++)
        fData[i][j] = (float) GetRandom(100, 1000);

PEvset (hWndPE, PEP_faYDATA, fData, 80); /* 4subset x 20points = 80 elements */

/* 4 tab delimited subset labels initialized above */
PEszset (hWndPE, PEP_szaSUBSETLABELS, szSubsetlabels, 4);

/* another way to set string property arrays */
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 0, "January");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 1, "February");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 2, "March");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 3, "April");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 4, "May");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 5, "June");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 6, "July");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 7, "August");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 8, "September");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 9, "October");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 10, "November");
PEvsetcell (hWndPE, PEP_szaPOINTLABELS, 11, "December");

/* reinitialize and reset image */
PEreinitialize (hWndPE);
PEresetimage (hWndPE, NULL, NULL);
```

## VBX Example

The following code is for a VBX Graph Object placed into a Visual Basic form. The code is from the Form\_Load event and the Graph Object has the following properties set:

Subsets = 4    Points = 12

```
    '** transfer some random data into object
    '** using VBX property to set data
    For s% = 0 To 3
        For p% = 0 To 11
            o% = ((s% * 12) + p%)
            PEGraph1.YData(o%) = (p% + 1) * 5 + (Rnd * (25))
        Next p%
    Next s%

    'DLL method of passing data, much faster when (Subsets * Points) > 500
    For s% = 0 To 3          {' 4 subsets}
        For p% = 0 To 11    {' 12 points}
            fData(s, p) = (5.0 + (Rnd * 25.0))
        Next p%
    Next s%
    result% = PEvset(PEGraph1, PEP_faYDATA, fData(0,0), 48);

    PEGraph1.SubsetLabels(0) = "Texas"      ' ** set Subset Labels **'
    PEGraph1.SubsetLabels(1) = "Florida"
    PEGraph1.SubsetLabels(2) = "Washington"
    PEGraph1.SubsetLabels(3) = "California"

    PEGraph1.PointLabels(0) = "January"    ' ** set Point Labels **'
    PEGraph1.PointLabels(1) = "February"
    PEGraph1.PointLabels(2) = "March"
    PEGraph1.PointLabels(3) = "April"
    PEGraph1.PointLabels(4) = "May"
    PEGraph1.PointLabels(5) = "June"
    PEGraph1.PointLabels(6) = "July"
    PEGraph1.PointLabels(7) = "August"
    PEGraph1.PointLabels(8) = "September"
    PEGraph1.PointLabels(9) = "October"
    PEGraph1.PointLabels(10) = "November"
    PEGraph1.PointLabels(11) = "December"

    PEGraph1.Refresh      ' ** must refresh after setting data **'
```

## DELPHI Example

The following code is for a VBX Graph Object placed into a Delphi form. The code is from the FormShow event and the Graph Object has the following properties set:

Subsets = 4    Points = 12

```
procedure TForm2.FormShow(Sender: TObject);
begin

  {'** Make Random Data for Graph Object }
  { VBX method of passing data to ProEssentials }
  { 4 subsets by 12 points = total points }
  {For s := 0 To 3 Do Begin
    For p := 0 To 11 Do Begin
      o := ((s * PEGraph1.Points) + p);
      PEGraph1.YData[o] := (5.0 + (Random * 25.0));
    end;
  end;}

  { DLL method of passing data, faster when (Subsets * Points) > 500 }
  For s := 0 To 3 Do          {' 4 subsets}
    For p := 0 To 11 Do      {' 12 points}
      fData[s, p] := (5.0 + (Random * 25.0));
  PEvset(PEGraph1.hObject, PEP_faYDATA, @fData, 48);

  {'** Set SubsetLabels property array for 4 subsets **'}
  PEGraph1.SubsetLabels[0] := 'Texas';
  PEGraph1.SubsetLabels[1] := 'Florida';
  PEGraph1.SubsetLabels[2] := 'Washington';
  PEGraph1.SubsetLabels[3] := 'California';

  {'** Set PointLabels property array for 12 points **'}
  PEGraph1.PointLabels[0] := 'January';
  PEGraph1.PointLabels[1] := 'February';
  PEGraph1.PointLabels[2] := 'March';
  PEGraph1.PointLabels[3] := 'April';
  PEGraph1.PointLabels[4] := 'May';
  PEGraph1.PointLabels[5] := 'June';
  PEGraph1.PointLabels[6] := 'July';
  PEGraph1.PointLabels[7] := 'August';
  PEGraph1.PointLabels[8] := 'September';
  PEGraph1.PointLabels[9] := 'October';
  PEGraph1.PointLabels[10] := 'November';
  PEGraph1.PointLabels[11] := 'December';

  {'** Always Refresh (PEactions:=0)after giving an object
    new data, YData, XData, SubsetLabels, PointLabels.}

  PEGraph1.PEactions := 0; (PEreinitialize)

end;
```



## 4. Conclusion

The ProEssentials offer many unique and innovative features which shape the ProEssentials into a high quality presentation charting solution. If you are working on an internal or vertical market application, you can utilize the ProEssentials functionality royalty free.

Whether you want quality data-driven images or a complete turn-key graphic presentation/exportation sub-system, the ProEssentials will elevate your current and future applications to a new level.

Gigasoft will greatly appreciate your business, and Gigasoft is committed to customer service and orientation. Please let us be your partner in producing the highest quality software possible.

**The ProEssentials are available for \$249.**

**Shipping is \$12 Overnight, \$8 Second Day, \$5 Ground. We take Mastercard & Visa.**

**Gigasoft, Inc. 696 Lantana Dr. Keller, TX 76248 (817) 431-8470 Fax: (817) 431-9860**

