

USER GUIDE

*free!*

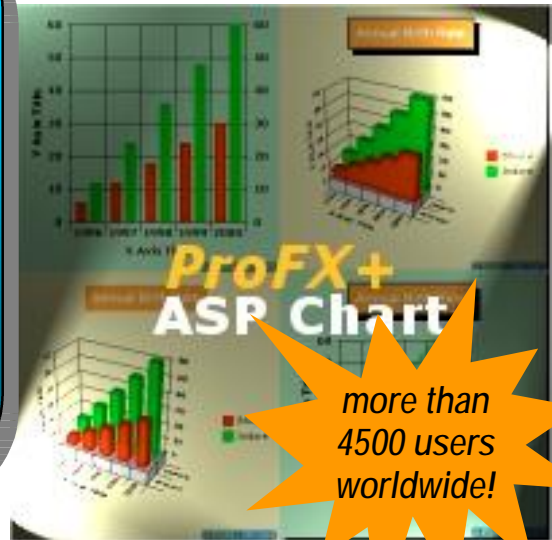
# ProFX+ ASP Chart Component



**APPEAL:** Developing ProFX+ ASP Chart Component has grown a time consuming job for me over the past years. I have to add new properties and methods and answer tens of emails every month, and at the same time continue to improve and test the software. I firmly believe that software for private use should be free and no intention to change this product into a crippled or commercial version in the future. Still, I need to make a living and income to continue this work.

So, I am asking you to download this software, try it, and if you like it and can spare a few bucks, please come back and donate. This money will be used only to cover the costs of ProFX+ ASP Chart Component developing and testing, and its absolutely voluntary! For donations you can contact me at:

[maresh\\_jo@hotmail.com](mailto:maresh_jo@hotmail.com) or [codefoundry@hotmail.com](mailto:codefoundry@hotmail.com)  
With Regards - Mahesh Joshi, Bhpooal [INDIA]



*The Ultimate  
Dynamic Charting  
Component for  
Active Server Pages*

*And what else,  
with 50 properties  
and 18 methods,  
its absolutely  
free!*

When publishing statistical information, web-sites always suffers lack of visitors due to tabular representation of figures. PROFX+ is a simple, easy to use Active Server Page Chart Component which provides 2D and 3D dynamic charting functionality for such web-sites. The component can be installed on a machine, running either Microsoft Internet Information Server (IIS) or Microsoft Personal Web Server (PWS). The component provides an in-built class "FXChart" which generates charts. The chart is generated and sent to the client browser as high quality JPEG or BMP image. To use it, first create a .ASP file which will be the target of the <img src=> tag of the .HTM or .ASP file which is requesting the chart. It provides selection of various types of charts like Bar, Line, Pie, Area and Step. The component is completely customizable and gives full control over itself. Some of its customizable features are:

- § Height and Width of chart
- § Chart Title font and font size selection
- § Chart background color selection (any RGB value)
- § Chart and Title Bar shadow control
- § Chart legends position and on/off control

Once the properties of the component are defined, SendChart method should be the final call, to generate the chart. The component has various properties and methods and setting them wrongly does not produce any error due to its underlying nature. Infect, an empty box without any chart is a coding error. For more details about component's properties, methods and practical examples, please go through the accompanied ProFXEx.HTM and CH?DBar.ASP, CH?DLine.ASP files. Following is a detailed description of the properties and methods of the PROFX+ ASP Chart component.

## WHAT IS NEW and VERSION HISTORY

Changes in ver 2.3 from ver 2.2 (appears in blue)

- Property to display data values on chart is added.
- Other Properties and Methods related to displaying the data values are also added.
- Method to change X and Y Axis Grid Line Color is added.
- Faster JPG engine.

Changes in ver 2.2 from ver 2.1

- § Property to change the Chart Title background is added.
- § Property to hide the second Y axis scale which appears at the right hand side of the chart.
- § Method for saving generated chart to local server path.
- § Method for creating edge border to columns of 2D & 3D bar chart.

Changes in ver 2.1 from ver 2

- § Ten new properties added, which all are related to chart Axis Scaling.

Changes in ver 2 from ver 1.2

- § This version is now only limited to IIS installations on Windows 2000/NT/XP platforms. There is no PWS version.
- § The ProFX+ text, which appears at the bottom of every chart, is now removed.
- § Font properties for axis titles is added.
- § Font properties for labels is added.
- § A new Property SeriesStacking is added .
- § PastePicture Method is added.

Changes in ver 1.2 from ver 1.0 (there is no ver 1.1)

- § fixed a bug which caused the ver 1.0 to stop after year 2004.
- § chart rotation method is added.
- § legend positioning method is added.
- § series coloring method is added.
- § legend font and font size selection property is added.
- § A new Property ImageType is added.

**Note:** If you are new to ASP component programming, please note that properties should be equated with its value by placing a '=' symbol between the property name and value. Whereas, methods are inbound procedures and values should be passed to them as parameter (i.e. no '=' sign in between). Following is an example of a property and a method respectively:

FXChart. Height=5	.....	Property
FXChart. CurrentRow 5	.....	Method

## PROPERTIES AND METHODS AT A GLANCE

PROPERTIES			
<b>.Height</b>	<b>.Width</b>	<b>.TitleFont</b>	<b>.TitleFontSize</b>
<b>.LegendFont</b>	<b>.LegendFontSize</b>	<b>.ImageType</b>	<b>.ChartType</b>
<b>.SeriesStacking</b>	<b>.TitleBorderSize</b>	<b>.TitleShadow</b>	<b>.XTitle</b>
<b>.XTitleFont</b>	<b>.XTitleFontSize</b>	<b>.Ytitle</b>	<b>.YTitleFont</b>
<b>.YTitleFontSize</b>	<b>.ZTitle</b>	<b>.ZtitleFont</b>	<b>.ZTitleFontSize</b>
<b>.XLabelFont</b>	<b>.XLabelFontSize</b>	<b>.YlabelFont</b>	<b>.YLabelFontSize</b>
<b>.ZLabelFont</b>	<b>.ZLabelFontSize</b>	<b>.Legends</b>	<b>.Shadow</b>
<b>.FillRed</b>	<b>.FillGreen</b>	<b>.FillBlue</b>	<b>.TitleBackgroundR</b>
<b>.TitleBackgroundG</b>	<b>.TitleBackgroundB</b>	<b>.AutoScaling</b>	<b>.XAxisMin</b>
<b>.YAxisMin</b>	<b>.ZAxisMin</b>	<b>.XaxisMax</b>	<b>.YAxisMax</b>
<b>.ZAxisMax</b>	<b>.XMajorDivision</b>	<b>.YmajorDivision</b>	<b>.ZmajorDivision</b>
<b>.SecondYAxisScale</b>	<b>.ChartTitle</b>	<b>.ShowDataValues</b>	<b>.DataValueFont</b>
<b>.DataValueFontSize</b>	<b>.DataValuePosition</b>		

METHODS			
<b>.Rows</b>	<b>.Columns</b>	<b>.CurrentRow</b>	<b>.CurrentColumn</b>
<b>.PastePicture</b>	<b>.RowTitle</b>	<b>.ColumnTitle</b>	<b>.SetData</b>
<b>.Rotate</b>	<b>.LegendPosition</b>	<b>.SeriesColor</b>	<b>.SaveChart</b>
<b>.SendChart</b>	<b>.BarEdge</b>	<b>.DataValueColor</b>	<b>.TitleFontColor</b>
<b>.XAxisGridColor</b>	<b>.YAxisGridColor</b>		

## PROPERTIES

**FXChart.Height = <number>**

Defines the height of the chart in inches. If not specified, a random value will be used by the component.

**Example:** `FXChart.Height = 5`

**FXChart.Width = <number>**

Defines the width of the chart in inches. If not specified, a random value will be used by the component.

**Example:** `FXChart.Width = 5`

**FXChart.ChartTitle = <string>**

Title of the Chart which will be displayed on the top of the chart. If set to blank, no title will be displayed.

**Example:** `FXChart.ChartTitle = "ANNUAL BIRTH RATE"`

**FXChart.TitleFont = <string>**

Name of the font in which the Title of the chart will be displayed. If not specified, a random font will be used by the component.

**Example:** `FXChart.TitleFont = "Arial"`

**FXChart.TitleFontSize = <number>**

Size of the title font. If not specified, a random font size will be used by the component.

**Example:** `FXChart.TitleFontSize = 14`

**FXChart.LegendFont = <string>**

Name of the font in which the legends of the chart will be displayed. The specified font must be installed on the server on which this component is running. If not specified, a random font will be used by the component.

**Example:** `FXChart.LegendFont = "Verdana"`

**FXChart.LegendFontSize = <number>**

Size of the legend font. If not specified, a random font size will be used by the component.

**Example:** `FXChart.LegendFontSize = 8`

**FXChart.ImageType = <string>**

Type of the image which will be transferred to the client application. Can be "BMP" or "JPG". If not specified JPG is used. *Watch out, this property is different from ChartType property which defines the chart style.*

**Example:** FXChart. ImageType = "BMP"

**FXChart.ChartType = <number>**

Any number between 1 to 8 (included). Each number specifies different chart type. Following are the chart type and their number codes:

1 - 2D Bar Chart	2 - 2D Line Chart	3 - 2D Area Chart	4 - 2D Pie Chart
5 - 3D Bar Chart	6 - 3D Line Chart	7 - 3D Area Chart	8 - 3D Step Chart

If not specified, 2D Bar chart type will be used by the component.

**Example:** FXChart. ChartType = 5

**FXChart.SeriesStacking = <Boolean>**

This is a new property in ver 2. Whether Stack the series or not? Default is False i.e. if not specified series are not stacked.

**Example:** FXChart. SeriesStacking = True ' or False

**FXChart.TitleBorderStyle = <number>**

Any number between 0 to 4 (included). Each number specifies different title border style. Following are the chart type and their number codes:

0 - No border	1 - Single Border	2 - Double Border	3 - Thick Inner
4 - Thick Outer			

If not specified, no title border will be plotted.

**Example:** FXChart. TitleBorderStyle = 2

**FXChart.TitleShadow = <number>**

A shadow effect can be applied to the chart title. 1 is used to plot a shadow, 0 for no shadow. If not specified, no title shadow will be plotted.

**Example:** FXChart. TitleShadow = 1

**FXChart.XTitle = <string>**

Specifies the title which will be plotted on the X axis of the chart.

**Example:** FXChart. XTitle = "Years"

**FXChart.XTitleFont = <string>**

Name of the font in which the X axis title will be displayed.

**Example:** FXChart.XTitleFont = "Tahoma"

**FXChart.XTitleFontSize = <number>**

Size of the font in which the X axis title will be displayed.

**Example:** FXChart.XTitleFontSize = 12

**FXChart.YTitle = <string>**

Specifies the title which will be plotted on the Y axis of the chart.

**Example:** FXChart.YTitle = "Birth Rate"

**FXChart.YTitleFont = <string>**

Name of the font in which the Y axis title will be displayed.

**Example:** FXChart.YTitleFont = "Tahoma"

**FXChart.YTitleFontSize = <number>**

Size of the font in which the Y axis title will be displayed.

**Example:** FXChart.YTitleFontSize = 12

**FXChart.ZTitle = <string>**

Specifies the title which will be plotted on the Z axis of the chart.

**Example:** FXChart.ZTitle = "Districts"

**FXChart.ZTitleFont = <string>**

Name of the font in which the Z axis title will be displayed.

**Example:** FXChart.ZTitleFont = "Tahoma"

**FXChart.ZTitleFontSize = <number>**

Size of the font in which the Z axis title will be displayed.

**Example:** FXChart.ZTitleFontSize = 12

**FXChart.XLabelFont = <string>**

Name of the font in which the values of X axis labels will be displayed.

**Example:** FXChart.XLabelFont = "Tahoma"

**FXChart.XLabelFontSize = <number>**

Size of the font in which the X axis labels will be displayed.

**Example:** FXChart.XLabelFontSize = 12

**FXChart.YLabelFont = <string>**

Name of the font in which the values of Y axis labels will be displayed.

**Example:** FXChart.YLabelFont = "Tahoma"

**FXChart.YLabelFontSize = <number>**

Size of the font in which the Y axis labels will be displayed.

**Example:** FXChart.YLabelFontSize = 12

**FXChart.ZLabelFont = <string>**

Name of the font in which the values of Z axis labels will be displayed.

**Example:** FXChart.ZLabelFont = "Tahoma"

**FXChart.ZLabelFontSize = <number>**

Size of the font in which the Z axis labels will be displayed.

**Example:** FXChart.ZLabelFontSize = 12

**FXChart.Legends = <number>**

Legends of the chart can be set on or off. 1 is used to show legend, 0 for no legend. If not specified, no legend will be plotted.

**Example:** FXChart.Legends = 1

**FXChart.Shadow = <number>**

A shadow effect can be applied to the entire chart. 1 is used to plot a shadow, 0 for no shadow. If not specified, no chart shadow will be plotted.

**Example:** FXChart.Shadow = 1

### Setting Chart Background

The background of the chart or the chart canvas color can be changed using following three properties. A color is a combination of Red, Green and Blue colors. You can specify intensity of individual colors to define your own chart background by using any number between 0 to 255.

**FXChart.FillRed = <number>**

**FXChart.FillGreen = <number>**

**FXChart.FillBlue = <number>**

**Example:** `FXChart.FillRed = 255`  
`FXChart.FillGreen= 255`  
`FXChart.FillBlue = 255` ' will set a white background

## Setting Chart Title Background

The background of the chart title can be changed using following three properties. A color is a combination of Red, Green and Blue colors. You can specify intensity of individual colors to define your own chart title background by using any number between 0 to 255. By default all three properties are set to 0 and a value of (R,G,B)=0 is no color i.e. Black. So if you see a black box instead of the title box, set a suitable title background color.

```
FXChart.TitleBackgroundR = <number>
FXChart.TitleBackgroundG = <number>
FXChart.TitleBackgroundB = <number>
```

```
Example:    FXChart.TitleBackgroundR = 0
              FXChart.TitleBackgroundG = 0
              FXChart.TitleBackgroundB = 255
              ' will set a Blue title background
```

## Axis Scaling Properties introduced from version 2.1

Ten new properties were introduced from version 2.1, which all are related to Axis Scaling. By default the axis scaling of the chart is done automatically, but user can define his own scaling for all axis. To use them, first set the AutoScaling property to false.

**FXChart.AutoScaling = <Boolean>**

**Before setting any other scaling related properties first set this property to false.**

**Example:** `FXChart.AutoScaling = False` ' or `true`

```
FXChart.XAxisMin = <number>
FXChart.YAxisMin = <number>
FXChart.ZAxisMin = <number>
```

### Minimum value of Axis.

**Example:** `FXChart.XAxis.Min = 10`  
`FXChart.YAxis.Min = 10`  
`FXChart.ZAxis.Min = 10` ' will set a minimum value of 10 to all axis

```
FXChart.XAxisMax = <number>
FXChart.YAxisMax = <number>
FXChart.ZAxisMax = <number>
```

**Maximum value of Axis.**

```
Example:    FXChart.XAxisMax = 300
              FXChart.YAxisMax = 300    'will set a maximum value of
              FXChart.ZAxisMax = 300    '300 to all axis
```



```
FXChart.XMajorDivision = <number>
FXChart.YMajorDivision = <number>
FXChart.ZMajorDivision = <number>
```

**Value on which the given axis to be divided.**

**Example:** `FXChart.XMajorDivision = 10`  
`FXChart.YMajorDivision = 10`  
`FXChart.ZMajorDivision = 10`  
`' will divide all axis on 10`

**FXChart.SecondYAxisScale** = <Boolean>

Default is true. Set this property to false to remove the second Y axis scale which appears on the right hand side of the chart.

**Example:** `FXChart.SecondYAxisScale = False` or `true`

## FXChart.ShowDataValues = <Boolean>

Default is false. Set this property to true to display the data values on chart. Use other related properties and methods to customize the font, font size and font color of data values.

**Example:** `FXChart.ShowDataValues = True` ' or false

## FXChart.DataValueFont = <String>

Name of the font in which the data values will be displayed. If not specified but ShowDataValues property is set to true, by default “Arial” font will be used.

**Example:** `FXChart.DataValueFont = "Tahoma"`

### FXChart.DataValueFontSize = <Number>

Size of the font in which the data values will be displayed. If not specified but ShowDataValues property is set to true, by default 9 point “Arial” font will be used.

**Example:** `FXChart.DataValueFontSize = 12`

**FXChart.DataValuePosition = <number>**

Any number between 1 to 4 (included). Each number specifies different positioning of data values on the chart. Following are the available options:

1 - Above	2 - Top	3 - Center	4 - Bottom
-----------	---------	------------	------------

If not specified, but ShowDataValues property is set to true, data values will be displayed above the plotted bar, line etc.

**Example:** FXChart.DataValuePosition = 3  
'Display values in center of bar'

## METHODS

The component plots the chart on the basis of the data supplied to it in the form of Rows and Columns. Therefore before calling these methods make sure how many rows and columns of data you have. For example following table will be used demonstrate the use of these methods. The data is in 5 rows and 2 columns. The cells grayed are row and column headers.

	Bhopal	Indore
Year 1980	10	15
Year 1985	12	17
Year 1990	20	32
Year 1995	24	33
Year 2000	28	38

### **FXChart.Rows <number>**

Defines the total number of rows which will be plotted on X axis. If not specified a random value will be used.

**Example:** FXChart. Rows 5

### **FXChart.Columns <number>**

Defines the total number of columns which will be plotted on Y axis. If not specified a random value will be used.

**Example:** FXChart. Columns 5

### **FXChart.CurrentRow <number>**

Defines the active row of the chart. The number should not be more than total number of rows defined earlier using FXChart.Rows method.

**Example:** FXChart. CurrentRow 1

### **FXChart.CurrentColumn <number>**

Defines the active column of the chart. The number should not be more than total number of columns defined earlier using FXChart.Columns method.

**Example:** FXChart. CurrentColumn 1

### **FXChart.PastePicture <x, y, FileName>**

Paste any existing BMP/JPG/GIF picture on the chart at the coordinate (x,y). Only one picture can be pasted on the chart surface. You can call this method as many times as you want but the only last call to PastePicture will get effect. If the picture is not visible even after calling this method, either the picture file not present in given path or the file is corrupt.

**Example:** FXChart. PastePicture 10, 10, "C:\images\Logo.jpg"

**FXChart.RowTitle <string>**

Defines the title of a row. The title will be applied to the active row.

**Example:**    FXChart. CurrentRow 1  
                   FXChart. RowTitle "Bhopal "  
                   FXChart. CurrentRow 2  
                   FXChart. RowTitle "Indore"

**FXChart.ColumnTitle <string>**

Defines the title of a column. The title will be applied to the active column.

**Example:**    FXChart. CurrentColumn 1  
                   FXChart. RowTitle "Year 1980"  
                   FXChart. CurrentColumn 2  
                   FXChart. RowTitle "Year 1985"  
                   . . . . .

**FXChart.SetData <number>**

Assigns data value to current row and current column.

**Example:**    FXChart. CurrentRow 1  
                   FXChart. CurrentColumn 1  
                   FXChart. SetData 10  
                   FXChart. CurrentColumn 2  
                   FXChart. SetData 15  
                   . . . . .

**FXChart.Rotate <number>**

Applies only to 3D charts, rotates it to given degrees.

**Example:**    FXChart. Rotate 45       ' Rotate chart by 45 degrees.

**FXChart.LegendPosition <number>**

Positions the legend according to given number. Any number between 0 to 7 (included). Each number specifies different legend position. Default is left. Following are the legend positions and their number codes:

0 - Top Left	1 - Top	2 - Top Right	3 - Left
4 - Right	5 - Bottom Left	6 - Bottom	7 - Bottom Right

**Example:** FXChart. LegendPosition 6

**FXChart.SeriesColor <number, red, green, blue>**

Defines the color of given data series. The method is optional and if this method is not used ProFX+ will use best suited colors for data series. First parameter should be a valid series number and following three numbers defines red, green and blue values.

**Example:**    FXChart. SeriesColor 1, 127, 127, 127  
                   'set color of data series 1 to gray.

**FXChart.BarEdge <series, width, red, green, blue>**

Defines the color of border of given data series. Default is no border i.e. a border with a zero width. First parameter should be a valid series number, then the width of the border and followed by three numbers defines red, green and blue values of the border.

**Example:**     FXChart.BarEdge 1, 20, 0, 0, 0  
                  'set black color border around data series 1 bar.

**FXChart.SaveChart <FileName>**

From this version, the generated chart can be saved to local server path for further use. This method should be called just before the SendChart method.

**Example:** FXChart. SaveChart "C: \saved. j pg"

**FXChart.SendChart**

The final call to FXChart class of PRO FX+ chart component. The chart will be sent to the client browser against the requested image source as JPEG/BMP image.

**Example:** FXChart. SendChart

**FXChart.DataValueColor <red, green, blue>**

Color, in which the data values will be displayed on the chart. Default color is White.

**Example:**     FXChart. DataVal ueCol or 0, 255, 0  
                  'set data value color to blue

**FXChart.TitleFontColor <red, green, blue>**

Color, in which the title of the chart will be displayed. Default color is White.

**Example:**     FXChart. Ti tleFontCol or 255, 255, 0  
                  'set chart title color to yellow

**FXChart.XAxisGridColor <red, green, blue>**

Color, in which the X Axis grid lines will be plotted. Default color is Black.

**Example:**     FXChart. XAxi sGri dCol or 255, 0, 0  
                  'set X axis grid color to Red

**FXChart.YAxisGridColor <red, green, blue>**

Color, in which the Y Axis grid lines will be plotted. Default color is Black.

**Example:**     FXChart. YAxi sGri dCol or 0, 0, 255  
                  'set Y axis grid color to Blue

## EXAMPLE SCRIPTS

For more details about component's properties, methods and practical example please go through the accompanied ProFXEx.HTM and CH?DBar, CH?DLine, CH??Area, CH??Pie and CH??Step ASP files. The ASP files may not be using all the available properties and methods, but you can try them by your own. To test the example scripts, copy all the ASP and HTM files provided with ProFX+ to web server's webroot (in most cases c: \inetpub\wwwroot) and in the Internet Explorer's address field type local host/profxex. htm and press enter.

Following is the result of local host /ProFXEx. HTM when called from the browser.

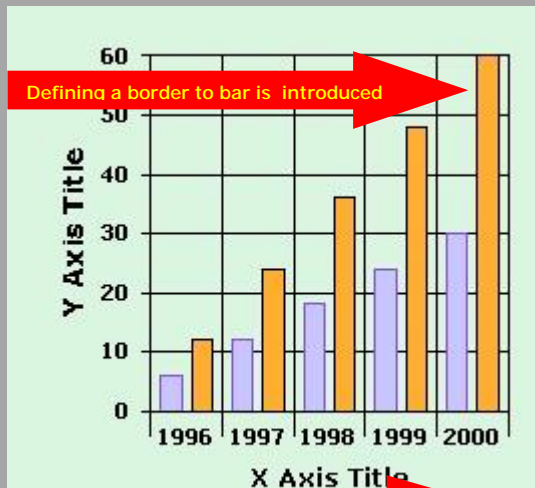
# PROFX+ ASP Chart Component

## Capability Demonstration Page

version  
**TWO.THREE**

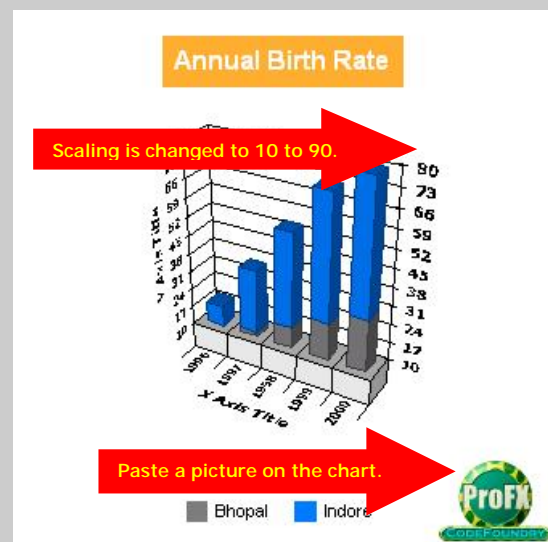
Why use figures to present statistics when a picture worth a thousand words!  
Enhance your site with dynamic charts using PROFX+ ASP Chart Component.  
Generate 2D and 3D Bar, Line, Area, Pie and Step charts with full customization.  
Believe it, following charts are generated dynamically on same data.

### PROFX+ 2D Bar Chart Example



### PROFX+ 3D Bar Chart Example

(Series Stacked and Picture Pasted - New Feature in Version 2)



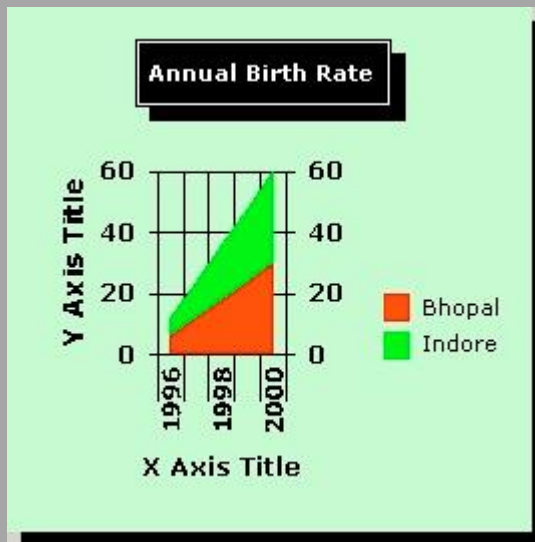
### PROFX+ 2D Line Chart Example



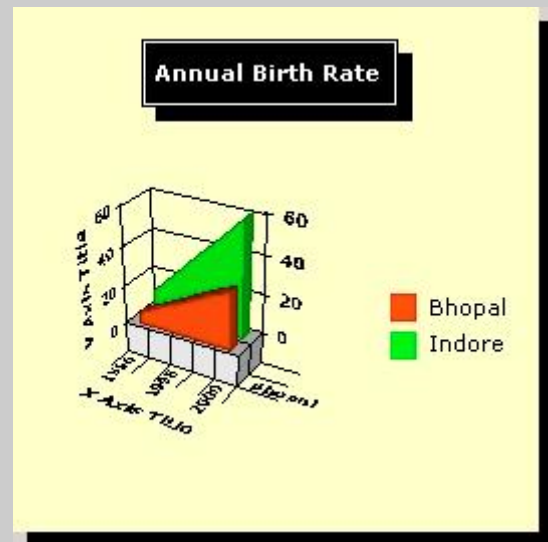
### PROFX+ 3D Line Chart Example



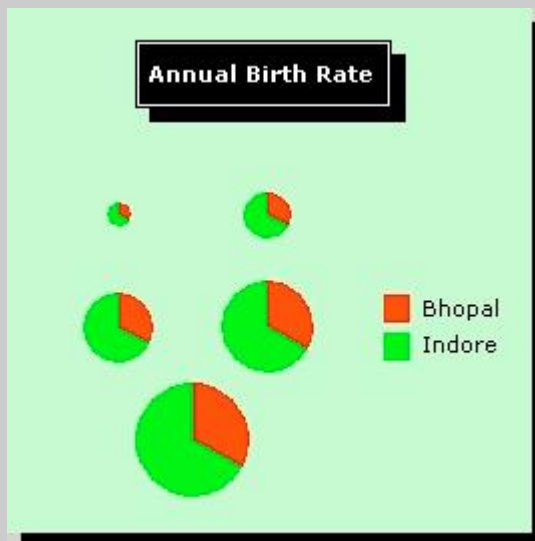
PROFX+ 2D Area Chart Example



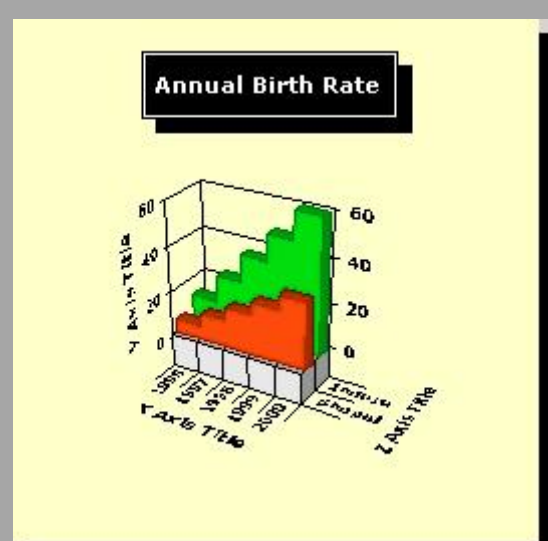
PROFX+ 3D Area Chart Example



PROFX+ 2D Pie Chart Example



PROFX+ 3D Step Chart Example



## MANUAL INSTALLATION

Although after running Setup, the ProFX+ ASP Charting Component will be installed properly, but still if does not work, you can check it. Following are the steps to check the proper installation:

- Go to Start/Programs/Administrative Tools/Component Services
- Double click on My Computer/COM+ Applications. You will get a list of installed COM+ Applications. Check for a component named as “ProFX”. If it is not there, its an installation problem. And if it so, you can install the component manually.

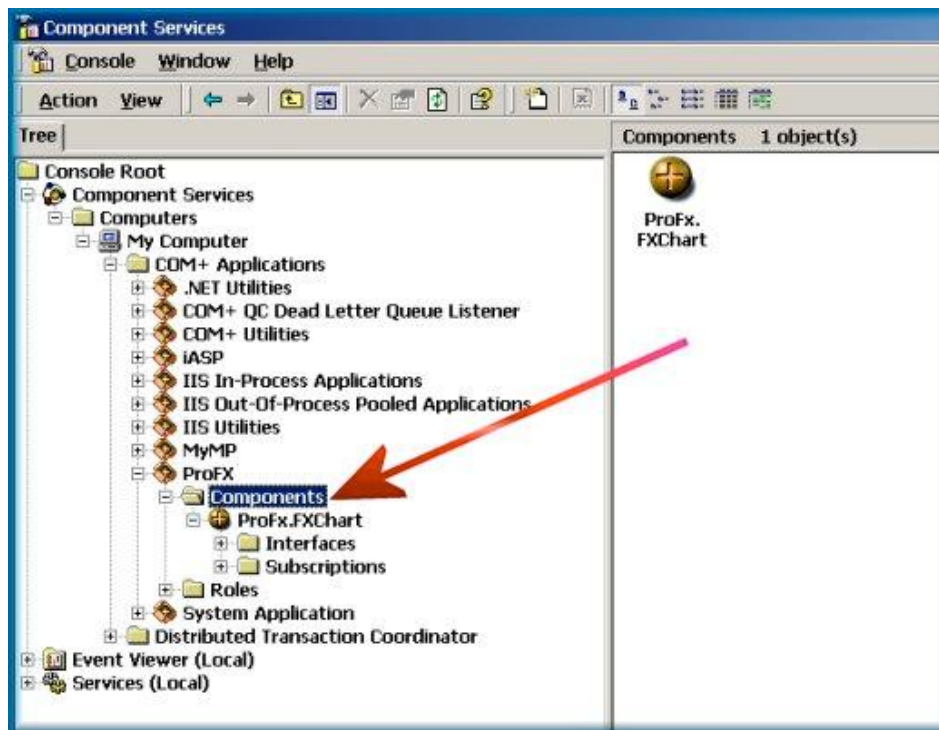
### Manual Installation Steps:

- Go to Start/Run and type CMD and press enter. You will get Command Prompt. The component resides within ProFX.DLL file and you can find it in SYSTEM or SYSTEM32 folder of your Windows installation (mostly WINNT). Give command `CD \WINNT\SYSTEM` or `CD \WINNT\SYSTEM32`, where the ProFX.DLL exists.
- Give command `REGSVR32 ProFX.DLL`. You will get an information dialog about registration of the DLL.
- Give command `EXIT`, so that the Command Prompt session ends. Go to Start/Programs/Administrative Tools/Component Services. Double click on My Computer/COM+ Applications. You will get a list of installed COM+ Applications. Right click on COM+ Applications on left pan and select New/Applications. Press Next and select “Create an empty application”.
- Type ProFX in name for the new application input box and select Server Application. Then press Next/Next/Finish.
- You will get a new COM+ Application as ProFX in right pan of Component Services Window.
- Double click on ProFX in left pan of window. You will get 2 nodes, Components and Roles. Double click on Components node so that it appears as opened folder.
- Right click on Components node. Select New/Component and press Next and select “Import component(s) that are already registered”. You will get a list of components which are already registered. Go through the list and select “ProFX.FXChart” and press Next/Finish.
- Restart the system.



On some systems (Windows 2003/XP) the DLLs and OCXs files get installed in C:\Documents and settings\myuser\windows folder. Move these files to C:\Windows\system32 folder and register them again. Also, you need to create COM+ object application as described above. You will also require to set some security settings manually. Set the security of the newly created COM+ application from Interactive user to System user.

Finally the Component Services Window should look like this:



This completes the manual installation of the component and now the component should work correctly. You can watch a complete video of above described process at CodeFoundry's ProFX+ download page.

**|| end of document ||**

ProFX+ ASP Charting Component is a royalty free product of:

CodeFoundry®  
Flat No.06, Sagar Garden Homes,  
Chunabhatti, Kolar Road, Bhopal (MP) - 462016 [INDIA]  
Phone: +091-0755-2424894

[codefoundry.netfirms.com](http://codefoundry.netfirms.com)

[codefoundry@hotmail.com](mailto:codefoundry@hotmail.com)