Ask a Question Write a Blog Post Login

Susmitha Susan Thomas

September 18, 2013 | 27 minute read

Drawing Graphical Charts with ABAP.

15 32 40,194

Follow

Like

RSS Feed

Many times a graphical chart output is required by the clients since a graph is much more appealing to the user and enables a better analysis of data. Some achieve this by exporting it to Excel and using the graph feature of Microsoft Excel. It is however quite simple to provide the functionality for building graphical charts in your ABAP report itself.

Including this feature in your ABAP report is an absolutely productive task since it could take your program to a much more professional level with relatively very little effort required from your side.

There are mainly two function modules that are being used to achieve this.

- Ø GFW_PRES_SHOW
- Ø GRAPH_MATRIX_3D

There is also a class called CL_GFW whose methods are dedicated to drawing graphs. However, most of the basic graph requirements can be achieved using the above function modules.

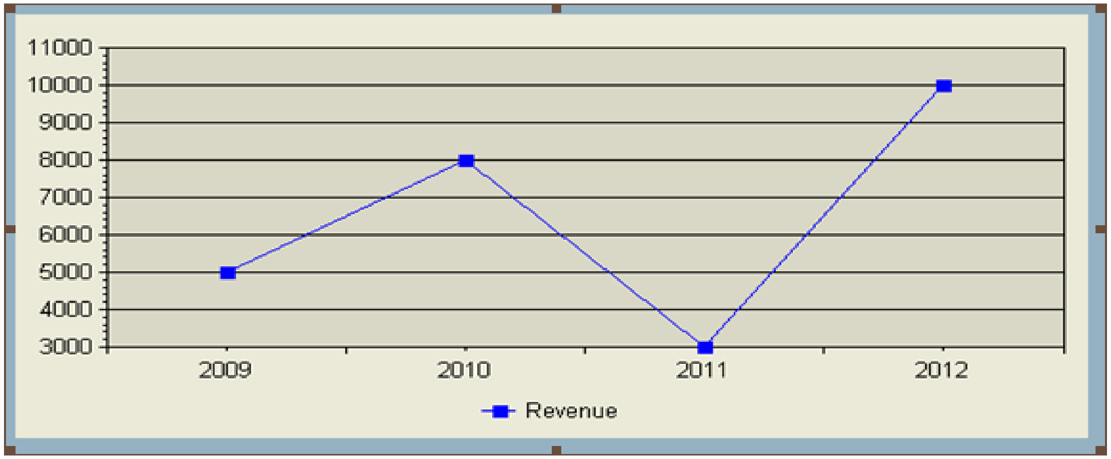
GFW_PRES_SHOW

Suppose you want to draw a line chart showing the revenue from a department over a few years.

These are the values for drawing the chart.

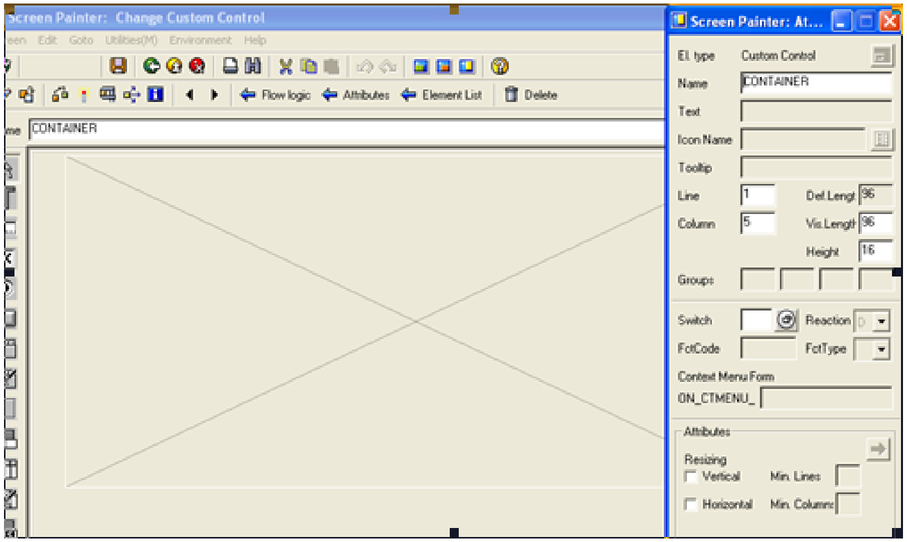
Year	Revenue
2009	5000
2010	8000
2011	3000
2012	10000

The output that you require would be :



To achieve this in ABAP

Pre-requisite : Create a custom screen say 100 and insert a custom control in it. Give it the name say ‘CONTAINER’



Name CONTAINER

El. type Custom Control
Name CONTAINER
Text
Icon Name
Tooltip
Line 1 Def.Lengt 96
Column 5 Vis.Lengtl 96
Height 16
Groups
Switch Reaction
FctCode FctType
Context Menu Form
ON_CTMENU_
□ Vertical Min. Lines
□ Horizontal Min. Columns

The graph can be easily drawn using the function GFW_PRES_SHOW.

CALL FUNCTION ‘GFW_PRES_SHOW’
EXPORTING

```
*           CONTAINER         =
*           TOP               =
*           LEFT              =
*           HEIGHT            =
*           WIDTH             =
PRESENTATION_TYPE    =
*           HEADER            =
*           ORIENTATION       = 1
*           PARENT            =
*           X_AXIS_TITLE      =
*           Y_AXIS_TITLE      =
*           FORMAT            =
*          IMPORTING
*           RETVAL            =
*           CONTENT_TYPE      =
*           CONTENT_LENGTH    =
TABLES
VALUES            =
COLUMN_TEXTS      =
*           ROW_LABELS        =
*           COLUMN_LABELS     =
*           CONTENT           =
*          EXCEPTIONS
*           ERROR_OCCURRED    = 1
*           OTHERS            = 2
.
```

The parameter **presentation type** determines the type of graph that needs to be drawn using the table. Given below are the values of the parameter presentation type corresponding to the different kinds of graphs.

1. Line charts – gfw_prestype_lines
2. Area Chart – gfw_prestype_area
3. Horizontal bar chart – gfw_prestype_horizontal_bars

4. Vertical bar chart – gfw_prestype_vertical_bars
5. Pie chart – gfw_prestype_pie_chart
6. Time axis chart. – gfw_prestype_time_axis

The important concept that needs to be understood while using this function module is the logic behind populating the input tables of the function module. Once you understand that, you can easily draw any kind of chart according to the requirement.

The two tables parameters **values** and **column_texts** are our messengers here.

If we consider the above example, column_texts contains the points along the x-axis. That is 2009, 2010, 2011.

The structure of the table column_texts is GPRTXT whichcontains just a single field COLTXT.

Append all the x-axis points into this table.

```
DATA :  X_TEXTS       TYPE TABLE OF GPRTXT WITH HEADER LINE.

X_TEXTS-COLTXT = ‘2009’.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2010’.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2011’.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2012’.
APPEND X_TEXTS.
```

The y-axis values corresponding to all x-axis co-ordinates is appended together as a single row in the table VALUES.

Values is a table of structure GPRVAL, which contains the field name for that graph and the values corresponding to the different x-co-ordinates.

ROWTXT	GFWXVAL	CHAR 40	0	GFW: First dimension (X value)

VAL1	GFWYVAL	FLTP	16	16	GFW: Second dimension (Y value)
VAL2	GFWYVAL	FLTP	16	16	GFW: Second dimension (Y value)
VAL3	GFWYVAL	FLTP	16	16	GFW: Second dimension (Y value)
VAL4	GFWYVAL	FLTP	16	16	GFW: Second dimension (Y value)
VAL5	GFWYVAL	FLTP	16	16	GFW: Second dimension (Y value)

You can enter up to 32 values, ie, upto 32 points on the x co-ordinate is allowed in the Texts table , the y value for each being provided in the Values table.

```
REPORT  ZGRAPHICS.

TYPE-POOLS: GFW.

DATA: Y_VALUES       TYPE TABLE OF GPRVAL WITH HEADER LINE,
X_TEXTS        TYPE TABLE OF GPRTXT WITH HEADER LINE.
data Ok_code like sy-ucomm.

REFRESH Y_VALUES.
REFRESH X_TEXTS.

Y_VALUES-ROWTXT = ‘Sales.
Y_VALUES-VAL1 = 5000.
Y_VALUES-VAL2 = 8000.
Y_VALUES-VAL3 = 3000.
Y_VALUES-VAL4 = 10000.
APPEND Y_VALUES.

X_TEXTS-COLTXT = ‘2009.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2010.
APPEND X_TEXTS.
```

```
X_TEXTS-COLTXT = ‘2011.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2012.
APPEND X_TEXTS.

call screen 100.
*&——————————————————————————*
*&      Module  STATUS_0100  OUTPUT
*&——————————————————————————
MODULE STATUS_0100 OUTPUT.
SET PF-STATUS ‘PF_100’.
SET TITLEBAR ‘GRAPHICS’.
*
CALL FUNCTION ‘GFW_PRES_SHOW’
EXPORTING
CONTAINER         = ‘CONTAINER’    “A screen with an empty container must be defined
PRESENTATION_TYPE = GFW_PRESTYPE_LINES
*       PRESENTATION_TYPE =  gfw_prestype_time_axis
*       PRESENTATION_TYPE = gfw_prestype_area
*       PRESENTATION_TYPE = gfw_prestype_horizontal_bars
TABLES
VALUES            = Y_VALUES
COLUMN_TEXTS      = X_TEXTS
EXCEPTIONS
ERROR_OCCURRED    = 1
OTHERS            = 2.

ENDMODULE.                ” STATUS_0100  OUTPUT
*&——————————————————————————*
*&      Module  USER_COMMAND_0100  INPUT
*&——————————————————————————
MODULE USER_COMMAND_0100 INPUT.
case ok_code.
when ‘BACK’.
leave PROGRAM.
```

```
endcase.

ENDMODULE
```

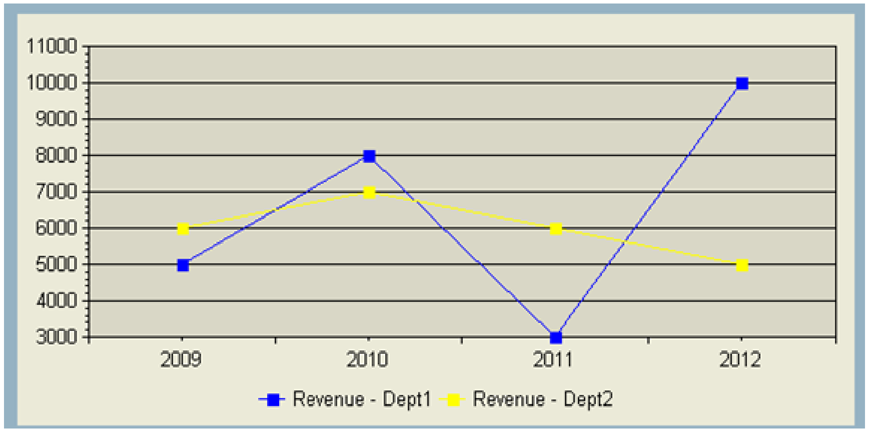
The output is the graph above.

Multiple Line Charts

Now suppose you want to compare the revenues of two different departments. You need a line chart, one for each department. To achieve this simply insert one more row to the y values tables.

```
DATA: Y_VALUES       TYPE TABLE OF GPRVAL WITH HEADER LINE,
X_TEXTS        TYPE TABLE OF GPRTXT WITH HEADER LINE.
data Ok_code like sy-ucomm.

REFRESH Y_VALUES.
REFRESH X_TEXTS.

Y_VALUES-ROWTXT = ‘Revenue – Dept1.
Y_VALUES-VAL1 = 5000.
Y_VALUES-VAL2 = 8000.
Y_VALUES-VAL3 = 3000.
Y_VALUES-VAL4 = 10000.
APPEND Y_VALUES.

Y_VALUES-ROWTXT = ‘Revenue – Dept2.
Y_VALUES-VAL1 = 6000.
Y_VALUES-VAL2 = 7000.
Y_VALUES-VAL3 = 6000.
Y_VALUES-VAL4 = 5000.
APPEND Y_VALUES.

X_TEXTS-COLTXT = ‘2009.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2010.
```

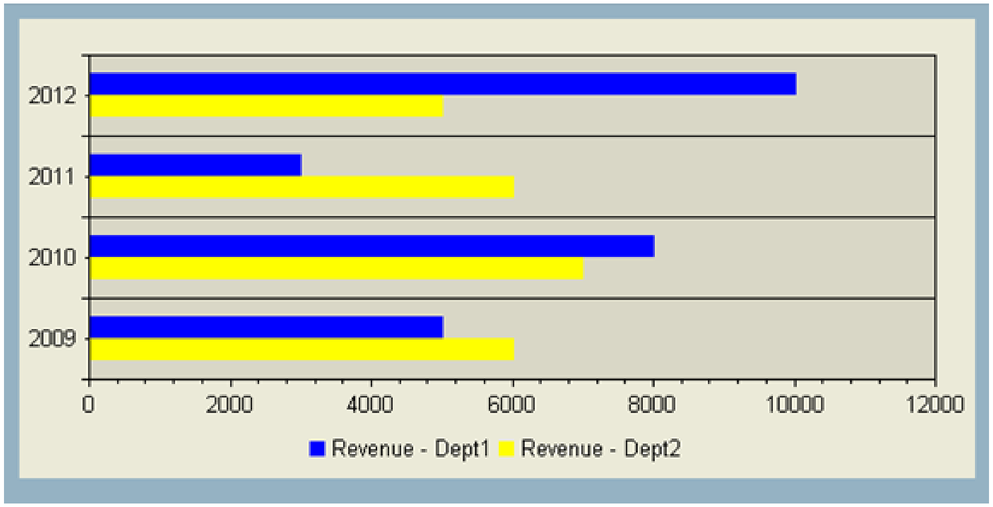
```
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2011.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘2012.
APPEND X_TEXTS.
```



Horizontal bar chart.

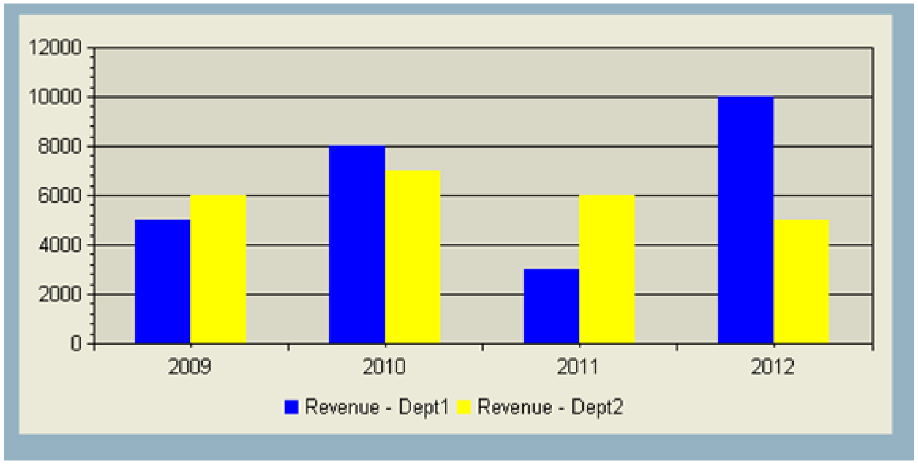
To view the same comparison in bar charts, just change the parameter passed to the presentation type parameter of the function module.

```
CALL FUNCTION ‘GFW_PRES_SHOW’
EXPORTING
CONTAINER        = ‘CONTAINER’

PRESENTATION_TYPE = gfw_prestype_horizontal_bars
TABLES
VALUES          = Y_VALUES
COLUMN_TEXTS     = X_TEXTS
EXCEPTIONS
ERROR_OCCURRED    = 1
OTHERS          = 2.
```



\Vertical Bar Chart

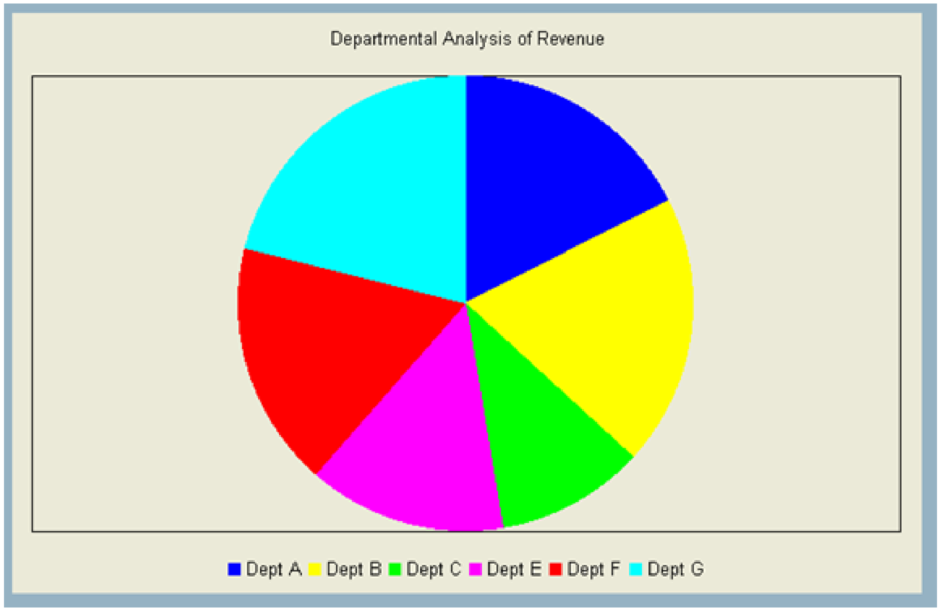
```
CALL FUNCTION ‘GFW_PRES_SHOW’
EXPORTING
CONTAINER        = ‘CONTAINER’

     PRESENTATION_TYPE = gfw_prestype_vertical_bars
TABLES
VALUES          = Y_VALUES
COLUMN_TEXTS      = X_TEXTS
EXCEPTIONS
ERROR_OCCURRED    = 1
OTHERS          = 2.
```



Pie Charts.

There is slight variation in the input table contents of pie charts for the very reason that the purpose of pie chart is different.

Suppose your requirement is to know the contribution of each department to the total revenue, the answer is a pie chart.



To draw pie chart, the different departments are added to the text table X_TEXTS (which will be the input to parameter **column_texts** of the function module) and the values, that is revenue, contributed by each department goes to the values table Y_VALUES (input table to parameter **values** of the table. There can be only one row for the Y_VALUES table for the pie chart.

```
DATA: Y_VALUES      TYPE TABLE OF GPRVAL WITH HEADER LINE,
X_TEXTS       TYPE TABLE OF GPRTXT WITH HEADER LINE.
data Ok_code like sy-ucomm.

REFRESH Y_VALUES.
REFRESH X_TEXTS.

X_TEXTS-COLTXT = ‘Dept A.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘Dept B.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘Dept C.
```

```
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘Dept E’.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘Dept F.
APPEND X_TEXTS.
X_TEXTS-COLTXT = ‘Dept G.
APPEND X_TEXTS.

Y_VALUES-VAL1 = 5000.
Y_VALUES-VAL2 = 5500.
Y_VALUES-VAL3 = 3000.
Y_VALUES-VAL4 = 4000.
Y_VALUES-VAL5 = 5000.
Y_VALUES-VAL6 = 6000.
Y_VALUES-VAL7 = 7000.
APPEND Y_VALUES.

CALL FUNCTION ‘GFW_PRES_SHOW’
EXPORTING
CONTAINER         = ‘CONTAINER’

       PRESENTATION_TYPE =  gfw_prestype_pie_chart
HEADER            = ‘Departmental Analysis of Revenue’
TABLES
VALUES            = Y_VALUES
COLUMN_TEXTS      = X_TEXTS
EXCEPTIONS
ERROR_OCCURRED    = 1
OTHERS            = 2.
```

GRAPH_3D / GRAPH_MATRIX* SERIES

The main function modules used in this series are

- Ø GRAPH_MATRIX
- Ø GRAPH_MATRIX_2D
- Ø GRAPH_MATRIX_3D
- Ø GRAPH_MATRIX_4D

All these function modules are similar. These FM may appear more complex, however, they offer more features than the previous FM, like sending graph as mail, setting window size etc. **An advantage in using these function modules is that you don’t have to create a container in screen painter to see the output.**

```
‘GRAPH_MATRIX_3D’ “Structure of 3D graphics (user-friendly version)

* EXPORTING

*   auto_cmd_1 = SPACE        ”          DO NOT USE

*   auto_cmd_2 = SPACE        ”          DO NOT USE

*   col1 = SPACE            ”          Column 1

*   col2 = SPACE            ”          Column 2

*   col3 = SPACE            ”          Column 3

*   col4 = SPACE            ”          Column 4

*   col5 = SPACE            ”          Column 5

*   col6 = SPACE            ”          Column 6

*   dim1 = SPACE             ”          Name of dimension 1

*   dim2 = SPACE             ”          Name of dimension 2

*   inbuf = SPACE            ”          DO NOT USE
*   inform = SPACE              ”              Allows dialog functions

*   mail_allow = SPACE          ”              Allows sending the graphic as mail in SAPoffice

*   pwdid = SPACE               ”              Dialog parameters

*   set_focus = ‘x’             ”              Set focus during reload

*   smfont = SPACE              ”              Use smaller fonts = ‘X’

*   so_contents = SPACE         ”              Subtitle of generated SAPoffice document

*   so_receiver = SPACE         ”              Recipient of generated SAPoffice object

*   so_send = SPACE             ”              Graphic storage in SAPoffice instead of display

*   so_title = SPACE            ”              Title of the generated SAPoffice document

*   stat = SPACE                ”              Dialog parameters

*   super = SPACE               ”              Dialog parameters

*   timer = SPACE               “

*   timer = SPACE               ”

*   titl = SPACE                “

*   titl = SPACE                ”              Title

*   valt = SPACE                ”              Scale name

*   valt = SPACE                “

*   wdid = SPACE                “
*   wdid = SPACE            ”           see WINID
*   winid = SPACE           ”           Dialog parameters
*   winid = SPACE           “
*   winpos = SPACE           ”           Window position
*   winpos = SPACE           “
*   winszx = ’50’          “
*   winszx = ’50’          ”          Window size in % / X
*   winszy = ’50’          ”          Window size in % / Y
*   winszy = ’50’          “
*   x_opt = SPACE           “
*   x_opt = SPACE           ”
*   notify = SPACE          ”           Activate ‘Save settings’
  IMPORTING
    b_key =                  ”          Pressed key
    b_typ =                  ”          Class for Business Graphics feedback
    cua_id =                 ”          Not yet
    mod_col =                 ”          Column of selected/modified object
    mod_row =                 ”          Line of selected/modified object
```

mod_val = ” New value of modified object

m_typ = ” Dialog parameters

rbuff = ” DO NOT USE

rwnid = ” Dialog parameters

TABLES

data = **”** Table with text fields and value fields

opts = **”** Options table

. ” GRAPH_MATRIX_3D

All the function modules in this series have more or less the same parameters with slight variations. Among the parameters, only the cols parameter, data table and opts table is compulsory. Rest is all additional features including setting the size of window (to change the default setting), sending graphic as mail etc.

The data table contains the input data for the graph while the ‘opts’ table defines the various settings that you require for your graph. You can pass an empty opts table when the graph will be displayed according to default settings.

The opts table will be a character table of the following structure.

data: begin of opts occurs 1,
c(80) type c,
end of opts.

Depending on the type of chart required, you need to populate this table. This parameter has a huge potential and almost any setting can be appended as a row in this table parameter.

Example :

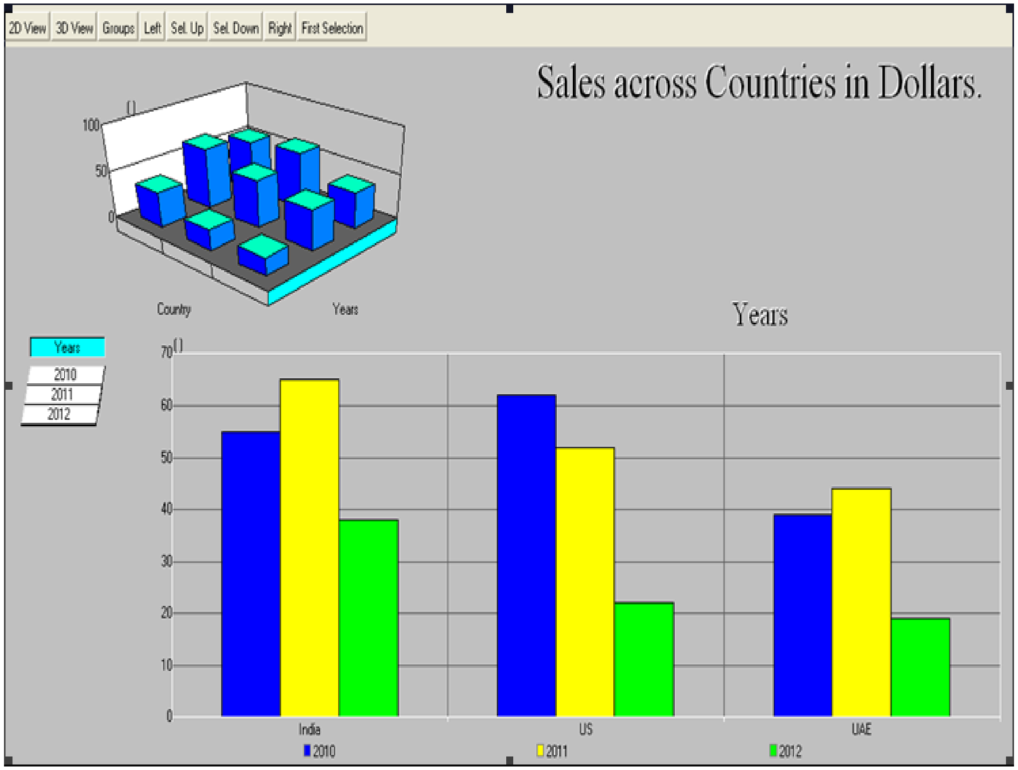
The following example outputs a 3D chart for the sales by country for three years.

It uses the FM GRAPH_MATRIX_3D

```
REPORT  ZGRAPHICS.
*
DATA: BEGIN OF ITAB_DATA OCCURS 0,
YEAR(15),
SALES1 TYPE I,
SALES2 TYPE I,
SALES3 TYPE I,
END OF ITAB_DATA,

BEGIN OF ITAB_OPT OCCURS 0,
OPTION(20),
END OF ITAB_OPT.

ITAB_DATA-year = ‘2010’.
ITAB_DATA-SALES1 = 55.
ITAB_DATA-SALES2 = 62.
ITAB_DATA-SALES3 = 39.
APPEND ITAB_DATA.

ITAB_DATA-YEAR = ‘2011.
ITAB_DATA-SALES1 = 65.
ITAB_DATA-SALES2 = 52.
ITAB_DATA-SALES3 = 44.
APPEND ITAB_DATA.

ITAB_DATA-YEAR = ‘2012.
ITAB_DATA-SALES1 = 38.
ITAB_DATA-SALES2 = 22.
ITAB_DATA-SALES3 = 19.
APPEND ITAB_DATA.
```

```
CALL FUNCTION ‘GRAPH_MATRIX_3D’
EXPORTING

      DIM1   = ‘Country’
      DIM2   = ‘Years’
       COL1   = ‘India’
       COL2   = ‘US’
       COL3   = ‘UAE’
       TITL   = ‘Sales across Countries in Dollars.’
    TABLES
       DATA   = ITAB_DATA
       OPTS   = ITAB_OPT
    EXCEPTIONS
       OTHERS = 1.
```

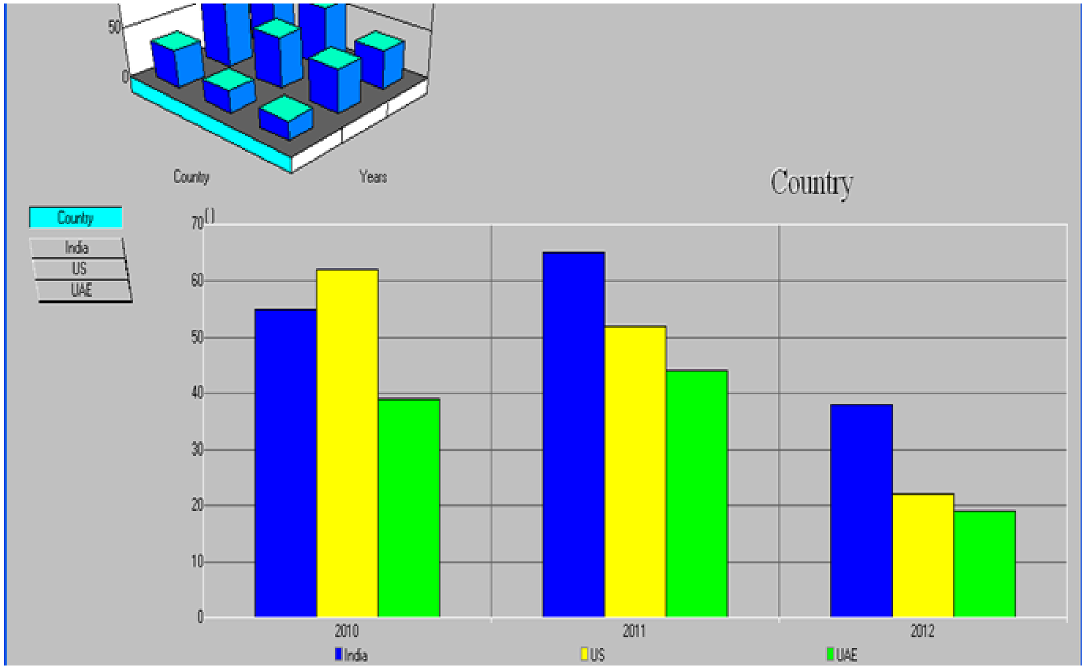
Output



This is the default output with an empty option table.

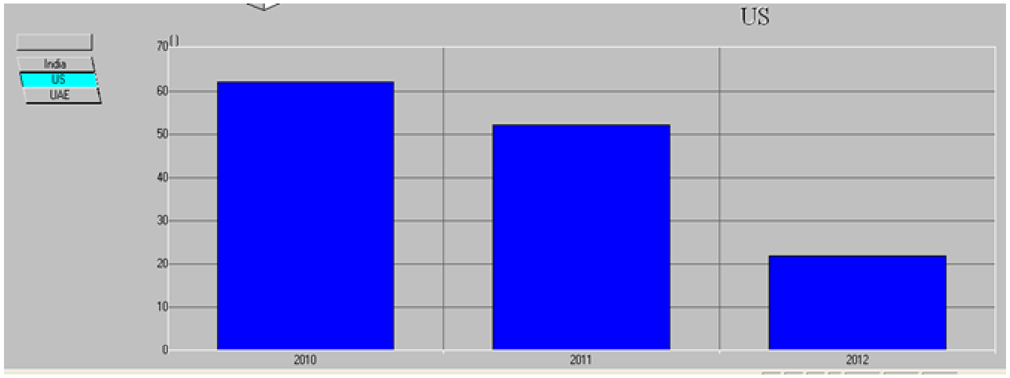
On the top left is the 3 D graph. All the three dimensions, namely Country, Year and Sales quantity can be seen together in the 3D graph. To get a bigger view of the 3 D graph, click on the ‘3D View’ button on the top.

The main graph in the lower right corner is the 2D vertical bar graph. You can change the view of the 2D graph by clicking on the button ‘Left’ or ‘Right’. The 2 D graph stack alternates between Years and Country on doing so.

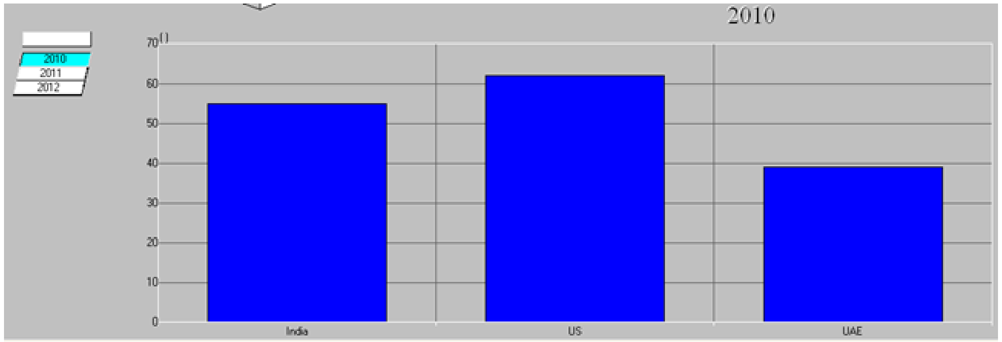


You can make the graph more specific by clicking on the Sel Down or Sel up Button. If 2D graph is grouped by country, it will show the graphs for each country one by one. You can also achieve this by navigating along the stack shown on the left side of the graph.

The sales output for the three years only for US is shown on navigating along the stack on the left side to US, or clicking the Sel Down Button Twice.



If the graph was grouped by Countries, we would have been able to see the graphs for each year for all countries.



First selection button alternates between the overall view and specialized view.

The Group Button gives a magnified view of 2D graph with each country/ year grouped as an independent graph.

This was the output for the FM with the default parameters in the OPT tab.

You can change the appearance of the 2D chart (the color or type of chart) by passing appropriate parameters to the opt table.

The following are the different values allowed in the opt parameter for 2D graphs.

Parameter for Type of 2D Graph

Parameter	Output
P2TYPE = VB	Vertical Bars
P2TYPE = VS	Stacked Vertical Bars
P2TYPE = HB	Horizontal Bars
P2TYPE = HS	Stacked Horizontal. Bars
P2TYPE = TD	Perspective Bars

Parameter	Output
P2TYPE = VT	Vertical Triangles
P2TYPE = ST	Stepped Lines
P2TYPE = MS	Stepped Areas
P2TYPE = LN	Lines
P2TYPE = SA	Stacked Areas
P2TYPE = MA	Shaded Areas
P2TYPE = PI	Pie Chart
P2TYPE = TP	Perspective Pie Chart
P2TYPE = PO	Polar Diagram
P2TYPE = PP	Relative Polar

The different type of 3D charts possible in the FM module are as below.

Parameter for Type of 3D Graph

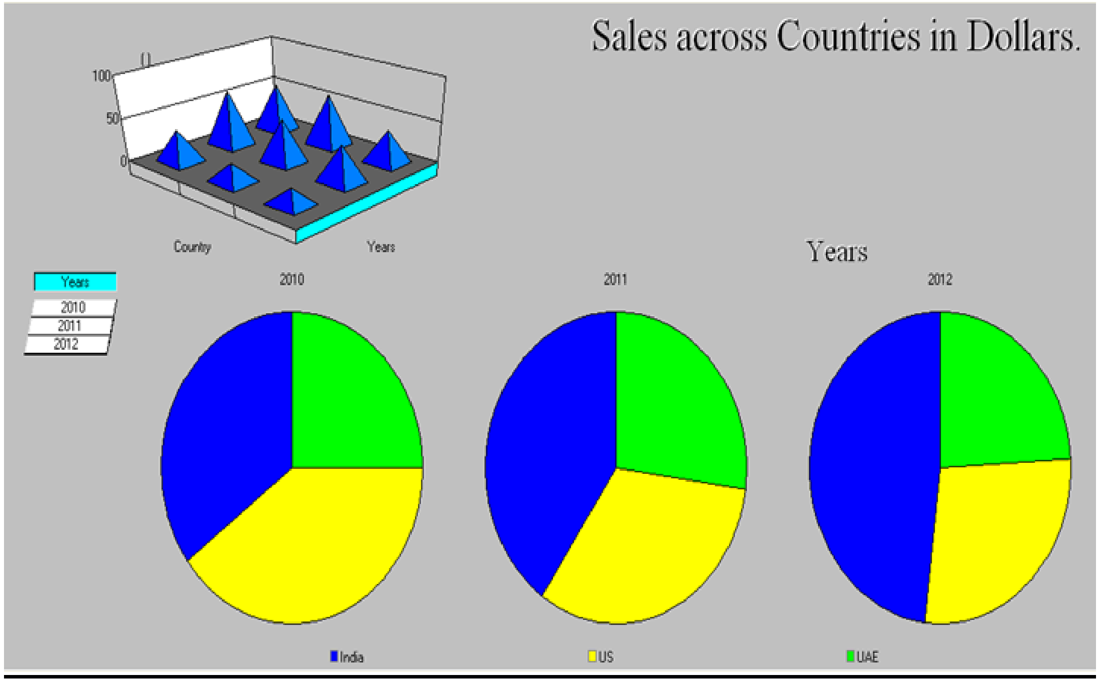
Parameter	Output
P3TYPE = TO	Towers
P3TYPE = PY	Pyramids
P3TYPE = ED	Walls
P3TYPE = WE	WEDGES
P3TYPE = LI	Strips
P3TYPE = NT	Surface

Now let us change the above program to output a Pie chart for the 2D graph, and a Pyramid chart for the 3D chart.

The only difference is that we populate the options table in addition to the above code. with the settings required.

```
itab_opt-option = ‘P3TYPE = PY’.  ” 3D Graph type – Pyramid
APPEND ITAB_OPT.
itab_opt-option = ‘P2TYPE = PI’.  ” 2D Graph type – Pie chart
APPEND ITAB_OPT.

*itab_opt-option = ‘FIFRST = 3D ‘.” First graph to display- 3D View

*APPEND ITAB_OPT.


CALL FUNCTION ‘GRAPH_MATRIX_3D’
EXPORTING
COL1   = ‘India’
COL2   = ‘US’
COL3   = ‘UAE’
TITL   = ‘Sales across Countries in Dollars.’
TABLES
DATA   = ITAB_DATA
OPTS   = ITAB_OPT
EXCEPTIONS
OTHERS = 1.
```

Output



We can also set the parameter for the first screen to be displayed. In the default settings, the Overview screen was displayed. Passing one of the following parameters, we can set a particular graph output to be displayed first. For example when we pass the parameter, ‘FIFRST = 3D’, it’s the magnified 3D view that is shown first on executing the transaction and we can see the other graphs by pressing the overview button.

Parameter for the First Graph to be displayed.

Parameter	Output
FIFRST = 2D	2D View (The output for **2D View** button on the Overview Screen)
FIFRST = 3D	3D View (The output for **3D view** button on the Overview Screen)
FIFRST = PU	Graph for the first component on the current stack is displayed first (In the example, 2010, ie, the output on clicking the Sel up/ Sel Down button on the Over view screen.)

Parameter	Output
FIFRST = GP	Groups (Output for Groups Button on the Overview Screen)
DTDORD = 2134	The graphs on the right stack will be displayed first. (In the example, the stack with Countries would be displayed, ie the output on Clicking the Right or Left button on the Overview Screen.)

The color of the graphs can also be changed with the following parameters

Parameter	Output
CLPALT = A	Blue
CLPALT = B	Yellow
CLPALT = C	Green
CLPALT = D	Gray
CLPALT = E	Dot Matrix Printer
CLPALT = F	PostScript

There are various other options available like the coloring scheme for the 3D graphs, background, 3 D effects, title, window size, which can be set by passing appropriate parameters in the table.

For more options available in this, refer the following link.

http://wiki.scn.sap.com/wiki/display/Snippets/Graph+Parameters

Note : The 2D graph options and 3D graph options can be used if you are calling GRAPH_MATRIX_3D FM, however only the options for 2D graph can be passed as parameters when GRAPH_MATRIX_2D FM is used.

GRAPH_MATRIX_4D.

With Graph_MATRIX_4D function module, I can add one more dimension to the graph. And then I can navigate the 2D and 3D graph for each value in the 4th Dimension by clicking on the Up and Down button.

The output looks similar to the output from the previous function with 3D graph at the top and 2D graph at the bottom

In the 3D graph, the fourth dimension is shown with colors.

In the previous program, if we want to give the sales comparison based on products also, I use this function module.

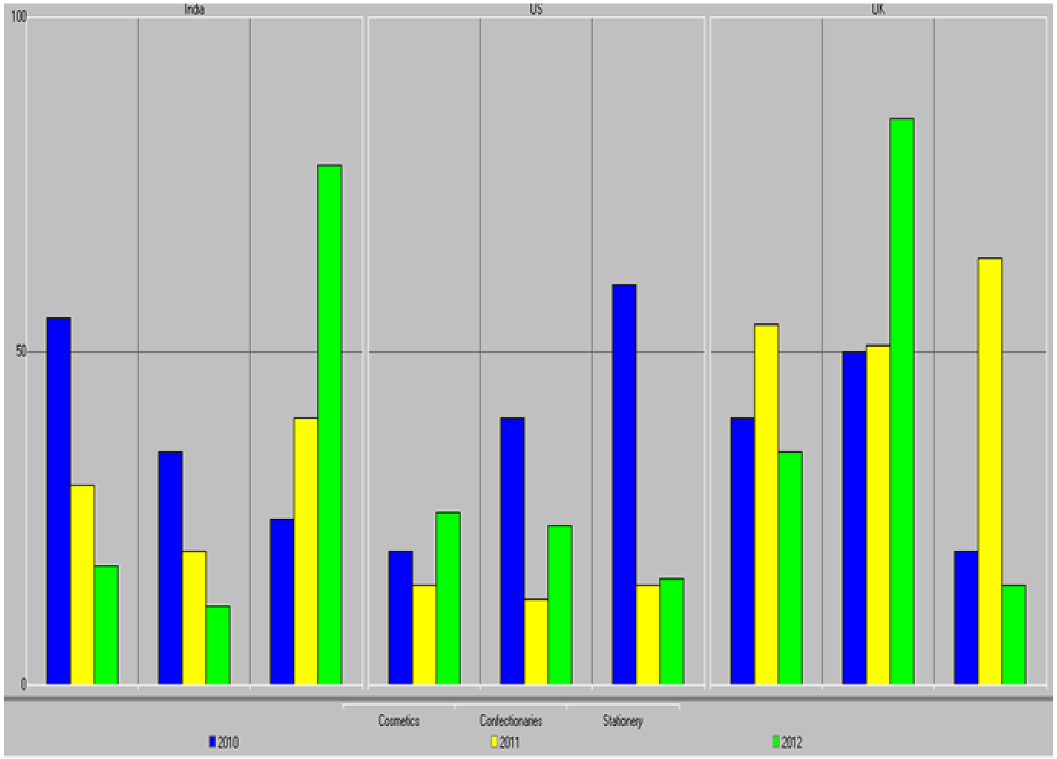
The output would be

Here in the 2D graph (bottom-right), the pzroduct Cosmetics is selected on the stack. The X axis, shows the Countries, the years are differentiated by the three colors. To see the graph for other products, just navigate along the products on the stack or click on the ‘Sel Up’ or ‘Sel Down’ Button above. It appears as if you are turning the pages of the book.

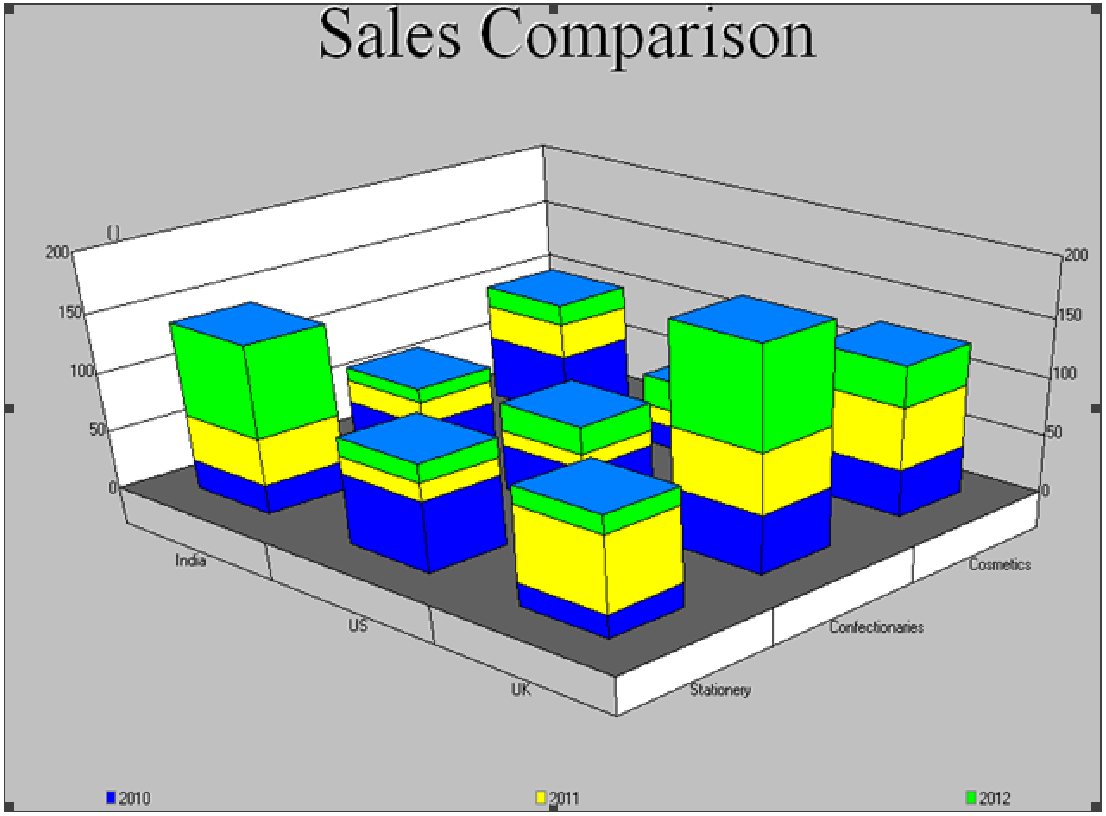
The y axis shows the sales for that particular combination.

The X axis and the color parameters can be alternated using the ‘Left’ or ‘Right’ Button at the top. On clicking them, the colors shows sales based on country and the years would be regulated along the x axis and vice-versa

The Groups button will show all the groups at on shot.



The 3D graph can be magnified on clicking the 3D button.



A

As can be seen, all the four dimensions are visible in this graph, using the x, y and z co-ordinates and the color code.

The colors on the graph can be enable or disabled by clicking the Stack On/Off Button.

Again, as was mention for the previous function module, using the appropriate parameters in the opts table, you can change the 2D graph to pie chart, horizontal bar chart etc. 3D chart type can also be changed according to the requirements. (Refer the parameter list provided above).

Now to get the above output, there are very few changes that need to be made to the function module compared to the 3D function.

When declaring the internal table, we obviously need to enter more values since there are four dimensions. However for the first dimension, a minimum of six fields must be entered in the internal table type, whether we require it or not.

```
types: BEGIN OF ITAB_type,
PROD(15),
IN10 TYPE I,
IN11 TYPE I,
IN12 TYPE I,
IN13 TYPE I,
IN14 TYPE I,
IN15 TYPE I,
US10 TYPE I,
US11 TYPE I,
US12 TYPE I,
US13 TYPE I,
US14 TYPE I,
US15 TYPE I,
UA10 TYPE I,
UA11 TYPE I,
UA12 TYPE I,
UA13 TYPE I,
UA14 TYPE I,
UA15 TYPE I,
END OF ITAB_type.

data : itab_data type table of itab_type with HEADER LINE,

BEGIN OF ITAB_OPT OCCURS 0,
OPTION(20),
END OF ITAB_OPT.

ITAB_DATA-prod = ‘Cosmetics’.
ITAB_DATA-IN10 = 55.
```

```
ITAB_DATA-IN11 = 20.
ITAB_DATA-IN12  = 40.
ITAB_DATA-US10  = 30.
ITAB_DATA-US11  = 15.
ITAB_DATA-US12  = 54.
ITAB_DATA-UA10  = 18.
ITAB_DATA-UA11  = 26.
ITAB_DATA-UA12  = 35.
APPEND ITAB_DATA.

ITAB_DATA-prod = ‘Confectionaries’.
ITAB_DATA-IN10 = 35.
ITAB_DATA-IN11 = 40.
ITAB_DATA-IN12  = 50.
ITAB_DATA-US10  = 20.
ITAB_DATA-US11  = 13.
ITAB_DATA-US12  = 51.
ITAB_DATA-UA10  = 12.
ITAB_DATA-UA11  = 24.
ITAB_DATA-UA12  = 85.
APPEND ITAB_DATA.

ITAB_DATA-prod = ‘Stationery’.
ITAB_DATA-IN10 = 25.
ITAB_DATA-IN11 = 60.
ITAB_DATA-IN12  = 20.
ITAB_DATA-US10  = 40.
ITAB_DATA-US11  = 15.
ITAB_DATA-US12  = 64.
ITAB_DATA-UA10  = 78.
ITAB_DATA-UA11  = 16.
ITAB_DATA-UA12  = 15.
APPEND ITAB_DATA.
```

```
* Grpah options
*itab_opt-option = ‘P3TYPE = PY’.  ” 3D Graph type – Pyramid
*APPEND ITAB_OPT.
*itab_opt-option = ‘P2TYPE = PI’.  ” 2D Graph type – Pie chart
*APPEND ITAB_OPT.
*itab_opt-option = ‘FIFRST = PU ‘. ” First graph to display – First graph on stack
*APPEND ITAB_OPT.
*itab_opt-option = ‘DTDORD = 2134’. ” The graphs on the right stack will be displayed first.
*APPEND ITAB_OPT.
*itab_opt-option = ‘FISTK3 = X’.
*APPEND ITAB_OPT.


CALL FUNCTION ‘GRAPH_MATRIX_4D’
EXPORTING
*   AUTO_CMD_1       = ‘ ‘
*   AUTO_CMD_2       = ‘ ‘
DIM1                  =  ‘Country ‘
DIM1_1                =  ‘India ‘
DIM1_2                =  ‘US’
DIM1_3                =  ‘UK ‘
*   DIM1_4          = ‘ ‘
*   DIM1_5          = ‘ ‘
*   DIM1_6          = ‘ ‘
DIM2                  =  ‘YEAR’
DIM2_1                =  ‘2010’
DIM2_2                =  ‘2011’
DIM2_3                =  ‘2012’
*   DIM2_4          = ‘ ‘
*   DIM2_5          = ‘ ‘
*   DIM2_6          = ‘ ‘
DIM3                  =  ‘Products’
*   MAIL_ALLOW       = ‘ ‘
```

```
TITL                = ‘ Sales Comparison’
TABLES
DATA                = ITAB_DATA
OPTS                = ITAB_OPT.
```

In the function module, DIM1, DIM2, and DIM3, defines the three dimensions of the graph, namely Country, Year and Products. The fourth dimension, Sales, is reflected from the Title of the Graph. The values for DIM1 (Country) are provided in the function modules as parameters, DIM1_1, DIM_2, DIM_3. The values in DIM2 (Year) is also similarly provided in the module. The Values in Dimension 3 (Products) is provided in the data table as the first column in each row appended.

To know about function module STAT_GRAPH_REF which is also used to draw graphs, refer the comment below from Ramesh T.

Drawing Graphical Charts with ABAP. | SCN

More advanced features can be achieved through the class CL_GFW.

Use transaction GRAL to explore into the various possibilities and features available in ABAP for graphical chart.

I hope that makes drawing graphs with ABAP complete and easily achievable. You can now confidently tell the client, ‘You say it, we got it!’

Enjoy drawing graphs on ABAP!

Assigned Tags

ABAP Development

abap

charts

gfs pres show

graph matrix 3d

graph matrix 4d

graphical

View more...

Similar Blog Posts

Data Analysis Tool

By Christoph Weyd Feb 20, 2023

SAP GUI Chart Engine usage & Trouble Shooting - Multiple Graphs in one Chart

By Santosh Baratam May 19, 2013

abap2xlsx – Business charts

By Alessandro Iannacci Dec 08, 2012

Related Questions

Is there anyway to make this code simpler?

By Former Member Oct 10, 2011

constants in smartform

By Former Member Mar 28, 2008

Table used in transaction cv03

By Former Member Aug 15, 2008

15 Comments

You must be Logged on to comment or reply to a post.

Shruti D

September 18, 2013 at 8:55 am

Very informative document. Thanks for sharing but the texts after graph comparison picture are truncated at right side.

Regards,

Shruti

Like 0 | Share

Former Member M

September 18, 2013 at 9:21 am

It is hidden, not truncated.

For example, click on line "The graphs on the right stack will be displayed first" and then use Left Right arrow keys to see hidden text. Tested on IE and Firefox browsers.

Like 0 | Share

Susmitha Susan Thomas | Blog Post Author
September 18, 2013 at 11:06 am

Thank you Shruti.

Thanks for the feedback regarding the image too. Yes, its hidden as Manish said. Anyways, I have reduced the size so that the entire image can be seen without have to click on it.

Like 0 | Share

Sreehari V Pillai
May 28, 2015 at 7:18 am

very useful doc Susmitha. 🙂

Like 0 | Share

Former Member
September 18, 2013 at 10:03 am

Nice.... but this concept already covered in many threads in SCN and step by step procedure also available in other sites...

Like 0 | Share

Susmitha Susan Thomas | Blog Post Author
September 18, 2013 at 12:10 pm

Thank you Ganesh,

Yes I did find many documents on Graphs, but didn't find much information on the population of internal table and the parameter values that need to be passed which forms the crux of the FMs. The features on the graph are also elaborate and require explanation. It took some time for me to master the whole thing and hence thought would compile it as a document to make things easier for others who would want to work on the same object.

Like 0 | Share

Ramesh T

September 25, 2013 at 10:00 am

Hi Susmitha,

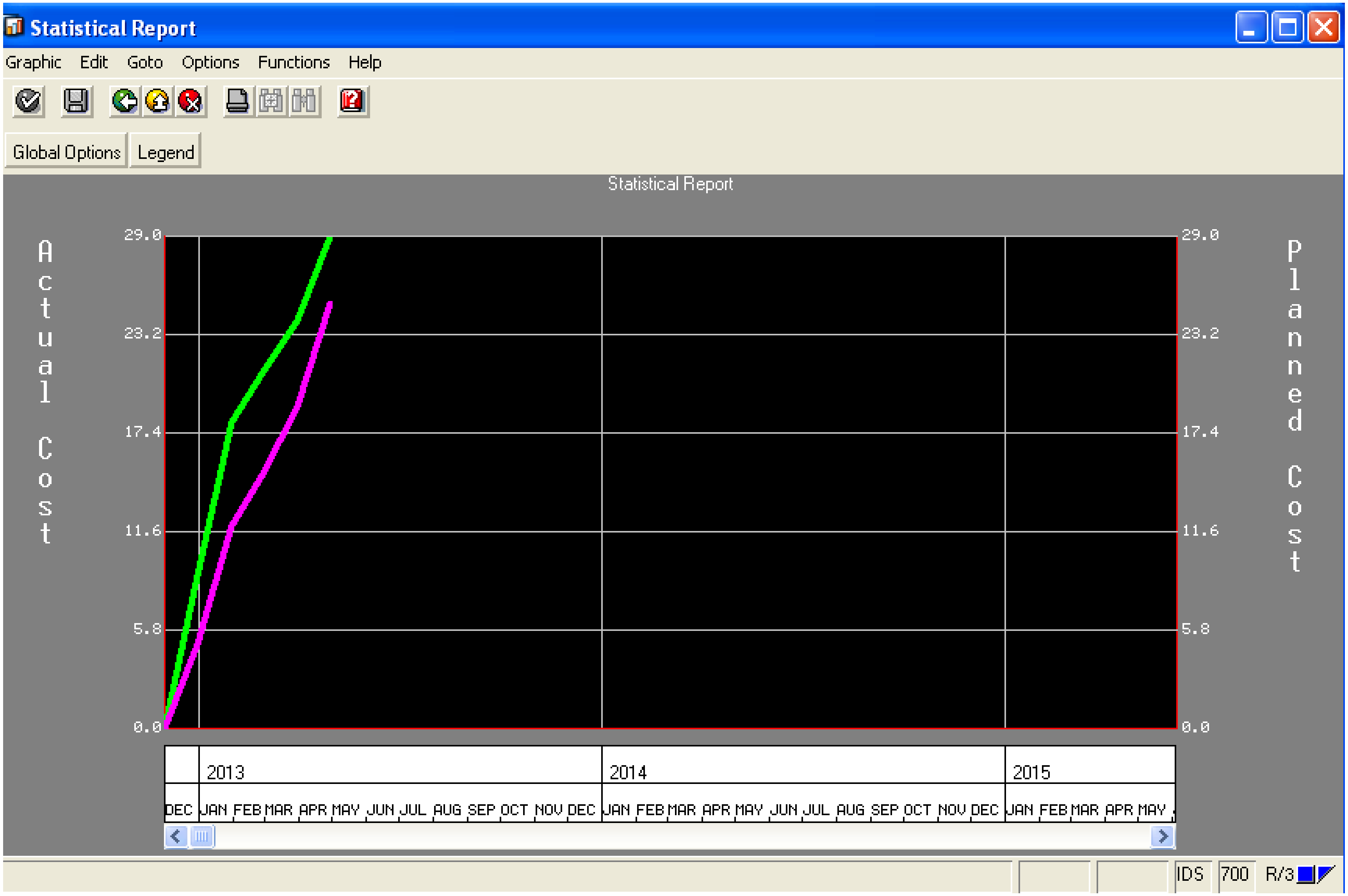
You can add this Function Module in your graphs

STAT_GRAPH_REF

```
*Structure Declarations
DATA:  BEGIN OF graf_par,
    dmode    VALUE '1',
    einheittext(6),
    gridtype VALUE '2',
    title    LIKE sy-title,
    ttext    LIKE strucr-kurztext,
    tunit    VALUE '6',
    utext    LIKE sy-title,
    valt(10),
    winpos(1) TYPE n VALUE  5,
    winszx(2) TYPE n VALUE 85,
    winszy(2) TYPE n VALUE 75,
  END OF graf_par,
*Internal Table Declaration
  BEGIN OF grafdata OCCURS 0,
    f TYPE i,
  END OF grafdata,

  BEGIN OF grafopts OCCURS 0,
    t(40),
  END OF grafopts,

  BEGIN OF graftime OCCURS 0,
    tick  TYPE d,
  END OF graftime.

*Start of selection*
START-OF-SELECTION.
* Append values to the Internal table for Title, Color, Text and Thickness
graf_par-title = 'Statistical Report'.
graf_par-ttext = 'Statistical Report'.
REFRESH grafopts.
grafopts-t = '$'.
APPEND grafopts.
grafopts-t = 'COLOR='.
REPLACE space WITH '4' INTO  grafopts-t.
APPEND grafopts.
grafopts-t = 'THICK=3'.
APPEND grafopts.
grafopts-t = 'LTEXT='.
Replace space with 'Actual Cost' into grafopts-t.
APPEND grafopts.
grafopts-t = 'C_ART=0'.
APPEND grafopts.  "Stützstellen

grafopts-t = '$'.
APPEND grafopts.
grafopts-t = 'COLOR='.
```

```
REPLACE space WITH '5' INTO  grafopts-t.
APPEND grafopts.
grafopts-t = 'THICK=3'.
APPEND grafopts.
grafopts-t = 'LTEXT='.
Replace space with 'Planned Cost' into grafopts-t.
APPEND grafopts.
grafopts-t = 'C_ART=0'.
APPEND grafopts.

*Append Date Values to the GRAFTIME Internal Table for X-axis.
graftime-tick = '20121201'.
APPEND graftime.
graftime-tick = '20130101'.
APPEND graftime.
graftime-tick = '20130201'.
APPEND graftime.
graftime-tick = '20130301'.
APPEND graftime.
graftime-tick = '20130401'.
APPEND graftime.
graftime-tick = '20130501'.
APPEND graftime.

* Append data to the GRAFDATA Internal table for Y-Axis
grafdata-f = 0.
APPEND grafdata.
grafdata-f = 9.
APPEND grafdata.
grafdata-f = 18.
APPEND grafdata.
grafdata-f = 21.
APPEND grafdata.
grafdata-f = 24.
APPEND grafdata.
```

```
grafdata-f = 29.
APPEND grafdata.

grafdata-f = 0.
APPEND grafdata.
grafdata-f = 5.
APPEND grafdata.
grafdata-f = 12.
APPEND grafdata.
grafdata-f = 15.
APPEND grafdata.
grafdata-f = 19.
APPEND grafdata.
grafdata-f = 25.
APPEND grafdata.

*Call STAT_GRAPH_REF FM for Statistical display
CALL FUNCTION 'STAT_GRAPH_REF'
  EXPORTING
    dmode      = graf_par-dmode
    gridtype   = graf_par-gridtype
    mail_allow = 'X'
    title      = graf_par-title
    ttext      = graf_par-ttext
    tunit      = graf_par-tunit
    utext      = graf_par-utext
    valt       = graf_par-valt
    winpos     = graf_par-winpos
    winszx     = graf_par-winszx
    winszy     = graf_par-winszy
  TABLES
    data       = grafdata
    opts       = grafopts
    vals       = graftime.
```


Statistical Report
Graphic
Edit
Goto
Options
Functions
Help
Global Options
Legend
Statistical Report
Actual Cost
Planned Cost
29.0
23.2
17.4
11.6
5.8
0.0
2013
2014
2015
DEC
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
JAN FEB MAR APR MAY
IDS
700
R/3

Like 0 | Share

Former Member

April 15, 2014 at 6:38 am

Hi Susmitha how we can send internal table data as a graph in mail body.

Like 0 | Share

Klaus Babl

April 15, 2014 at 7:14 am

Hi Susmitha,

thanks for your great document.

Weeks ago I was working with graphs using the classes CL_GUI_CHART_ENGINE and
LCL_DC_PRES (include GFW_DC_PRES), but your sample fm calls are looking very easy to handle and better for a fast solution.

Now still I have a question for showing the exact values in bars and pie charts. Did you come across a solution for that?

In your pie charts example you have 6 X values and 7 Y values, where the values 5000, 5500 and 6000 are very close together. It would be great, if the exact Y values could be shown there. Also for bars this would be a great idea.

I failed trying this with the classes mentioned above.

Keep on sharing your good studies and research on SCN.

Thanks again.

Klaus

Like 0 | Share

Susmitha Susan Thomas | Blog Post Author

June 2, 2014 at 1:14 pm

Thanks Gaurav Gautam!

Like 0 | Share

Yarnagula Sudhir

September 2, 2014 at 12:20 pm

How to display Values(Numbers) over every individual Towers in 3D Graph?

Hi Susmitha!

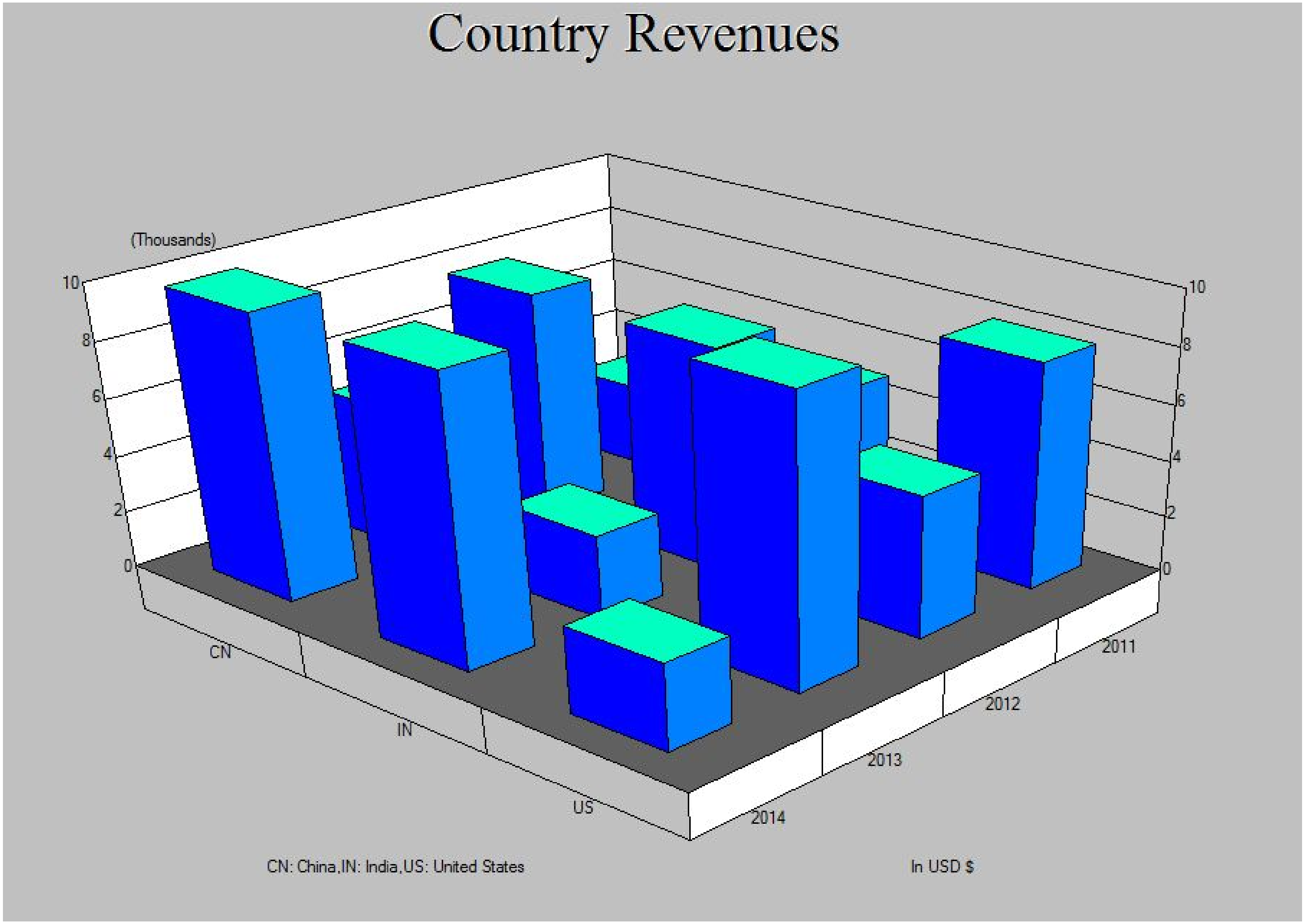
PFA below. Its regarding the screen shot of 3D Graph.

As seen in the picture that the values are not accurate to view. So its better to display values(numbers) on top of each tower in 3D Graph.

How to display that? Kindly suggest me any solution.

With Regards,

Yarnagula Sudhir.


Country Revenues
(Thousands)
10
8
6
4
2
0
CN
IN
US
2014
2013
2012
2011
CN: China,IN: India,US: United States
In USD $

Like 0 | Share

Miss Jyothi Sharma

September 28, 2014 at 5:24 pm

Hey Sushmitha, Thanks its really helpfull.

Like 0 | Share

Yarnagula Sudhir

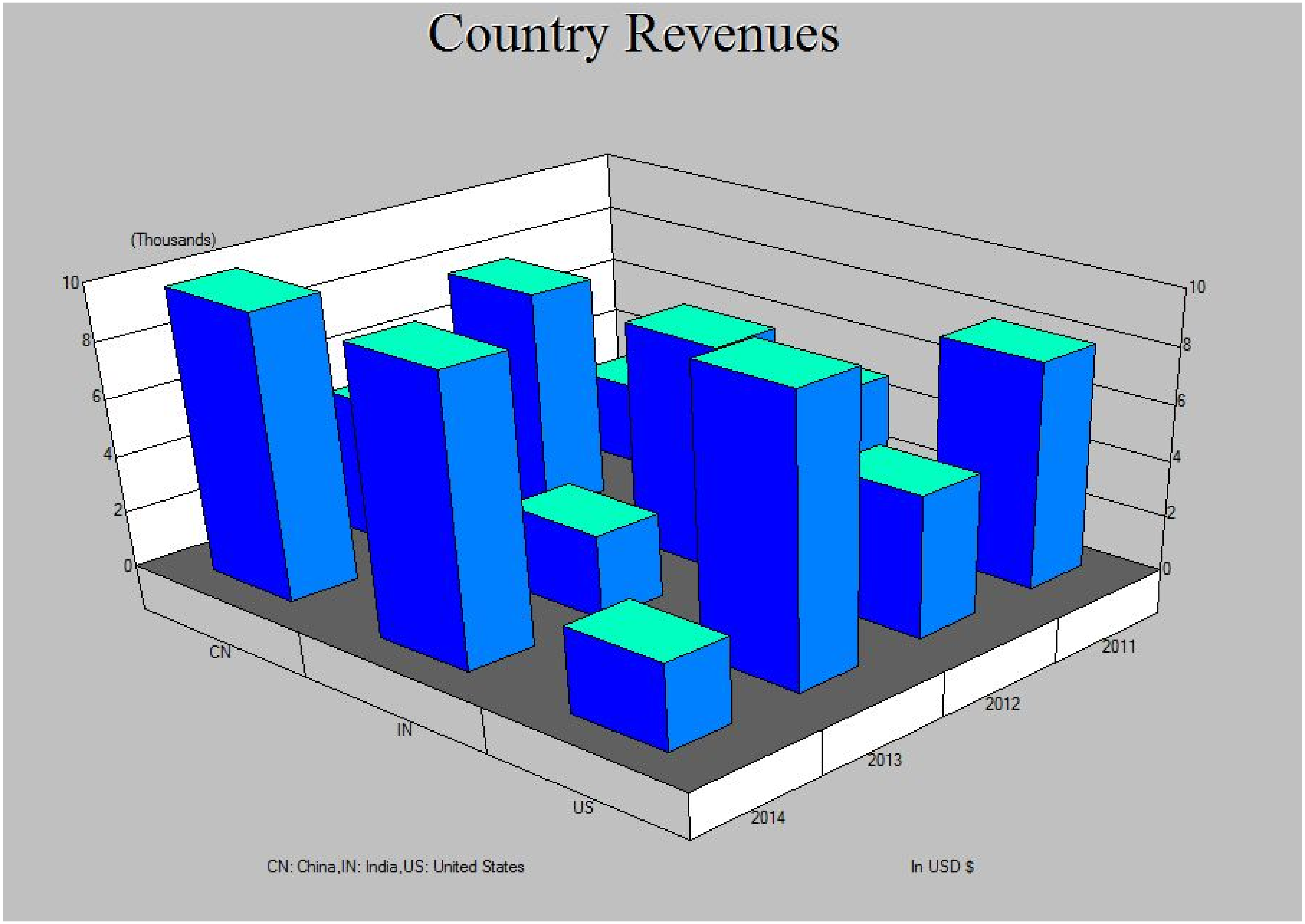
September 29, 2014 at 5:18 am

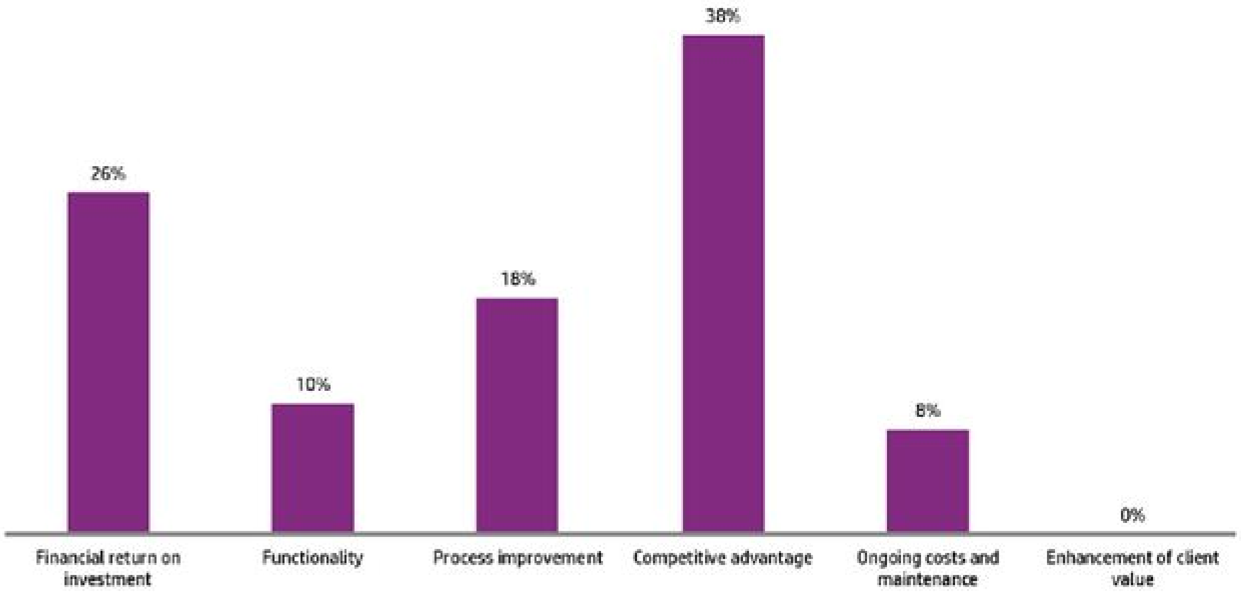
Hi Miss Jyothi Sharma!

Have you tried displaying Values on top of the Graph?

If so kindly let me know.

Note: Find the attachment Pictures for the reference.


Country Revenues
(Thousands)
10
8
6
4
2
0
10
8
6
4
2
0
CN
IN
US
2011
2012
2013
2014
CN: China,IN: India,US: United States
In USD $



With Regards,

Yarnagula Sudhir.

Like 0 | Share

Salvador Nava

February 24, 2016 at 3:36 pm

Thank you very much, this was so helpful!!

Like 0 | Share

Xiaoguang Li

June 23, 2022 at 5:07 am

Very informative document. Thanks

Like 0 | Share