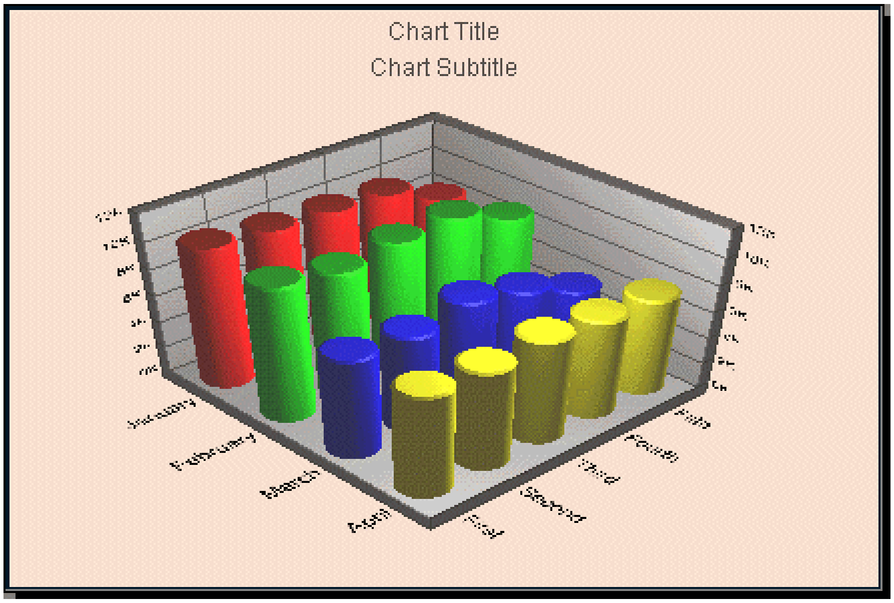
PRESENTATION GRAPHICS SDK
TECHICAL NOTES
BY THREE D GRAPHICS, INC.

PGSDK provides the tools you need to add professional charts and graphs to an application. The chart is drawn in your application's window or child window within the rectangular location your application indicates. Your application can draw any number of charts in any number of windows. It is limited only by available memory and Windows system resources.



- Basic Features
- Key Features
- Advanced Features
- What you get...

Also see these topics for more information:

- Creating a Chart
- User Selections
- Modifying Attributes
- Chart Attributes

- API Functions
- Compile and Link
- What else do you need to know?

Basic Features

Any Output DC: Drawing can be performed to any output DC -- screen, printer, memory and Windows metafile (WMF). This allows your application to copy a chart to the clipboard as a metafile picture or a bitmap. PGSDK supports True Windows copy and paste capability.

Selection Highlight: User-selected objects are easy to highlight. Minimal code is required to highlight a user selection and provide visual feedback to the user

Move and Resize Objects: Title, Subtitle, Footnote, Row Title, Column Title, Axis Titles, Frame, and Legend objects can be moved and sized dynamically by the user. When an object is selected and highlighted, eight resize handles appear on the object's bounding box. You can resize by clicking and dragging. PGSDK supplies sample code to show how easily this is done.

Annotations: You can draw rectangles, rounded rectangles, ellipses, polygons, lines, arrows, free hand lines, and text boxes.

Vertical/Horizontal Charts: 2D Bar, Line, and Area charts can be displayed vertically or horizontally.

Numeric Text Formats: Numeric text supports a wide variety of formats for miscellaneous values, dollar amounts, percentages, scientific notation, thousands, and millions.

Date & Time: PGSDK supports these data/time formats: "2/25/64", "Feb 25", "Tuesday, February 25, 1964", "18:56" (24 hr. time), "Feb 25, 64", "Feb 64", "Feb 25, 64 6:56 PM"

Object Poke Through: PGSDK supports the ability to find the next object even when objects are layered on top of each other. Just because an object is underneath doesn't mean it can't be selected.

Horizontal or Vertical Legends: Legends can be oriented horizontally or vertically or as rows and columns under your applications control.

Emphasize a Data Series: In a bar, line, or area chart a data series can be emphasized by drawing it as another chart type. For example, on a side by side bar chart your application can make one of the data series draw as a line or area riser.

Key Features

A Cross-Platform solution: PGSDK is the only library available for Windows (16&32 bit), Macintosh (680x0 & PowerMac), UNIX, **and** OS/2. With standard ANSI-C calls that work identically on all computers, you can be guaranteed of having great graphics regardless of the development platform, now or in the future.

Two different data models: PGSDK provides two unique ways of inserting data into a graph.

1. Data Callbacks: User-defined functions that the library calls when it draws a riser (basically, you provide the numbers only at the exact moment the library needs to chart them). Perfect for large and/or constantly changing data sets since it avoids constant update and storage issues.

2. Send Data Interface: A family of functions that allow you to "send" data to a chart where it is stored for you.

Graphs that actually draw in your own window: Unlike some competitors that draw your chart in another window and then attempt to "paste" the finished chart in your application, PGSDK creates charts in the Windows DC of your choice. This provides the fastest, cleanest and most intimate link possible.

On-screen editing and detection: PGSDK lets you click on **any** chart element and move or resize it. PGSDK provides XOR handles and highlights so that users get visual feedback on what items have been selected. Specialized feedback functions provide all the information needed to create data drill-down effects and provide an unparalleled degree of chart customization.

Advanced Special Effects: When you want to add some *sizzle* to your application, nothing quite compares to the effects offered by PGSDK. You can put bitmap or vector images on any chart element (choose if they stretch-to-fit or tile!). You can add drop-shadows or any of 16 types of gradient fills. You can even create Pictogram chart types for that "USA Today" look.

Left-to-Right and Up-Down Charts: PGSDK has Bi-polar, dual-Y, Horizontal/Vertical and Ascending/Descending controls. You can create special effects such as bar charts that draw from the top down or comparison charts that draw from the middle out (for example, boys on the left and girls on the right).

Fantastic 3D Charts! PGSDK offers complete customization of your 3D viewing parameters as well as 16 "preset" 3D Viewing Angles so you can provide your users a multitude of ways to see their data without fear of getting lost in 3D space.

AutoFit Font Technology: PGSDK has font technology that allows you to resize the chart in any manner without destroying its readability. Whether Big, Small, Short & Wide, Tall & Skinny, it will automatically select a font size that looks good so you get great looking charts every time.

Endless Customization: With PGSDK, you get individual control over the Pie Thickness, Location, Tilt Angle, Rotation Direction, Start Angle, Feeler Position and much more. You also get complete control over the text size and location. With every chart type, PGSDK provides a range of custom features.

Full Internationalization Support: If your plans include overseas sale or development, PGSDK is a worldwide charting tool. With features like automatic detection of currency and decimal symbols, easily editable resource strings and double-byte character support, this library is ready to go wherever you go.

Undo and Redo Support: PGSDK also provides a powerful undo/redo facility so your users can experiment without any fear of wasting time/destroying previous work. Choose from last-action or Anchor-style undo schemes.

RAMFILES: The Presentation Graphics SDK has the ability to load and save chart files directly into memory (called RAMFILES). This provides you with the ability to store frequently used chart templates in resources or embed your user's charts inside your own file format.

Year 2000 Compliant: The PGSDK does not manipulate, store, or transform any dates or date-related numbers, strings, or tags anywhere in its code. It cannot be affected in any way by date changes on a client machine. This compliance has been tested and verified at Three |D| Graphics.

Advanced Features

Open GL: Advanced 3D Rendering is an alternate form of drawing that is optionally done when a client calls DrawTheGraph(). Instead of platform-specific API calls (for example to the Windows GDI or Macintosh Toolbox), Advanced 3D uses the OpenGL graphics library to display complex 3D versions of charts with dynamic lighting, z-buffering, 3D text, true texture-mapping and other visual enhancements. In addition, OpenGL takes advantage of any available 3D hardware acceleration if the proper drivers are installed. Advanced 3D attributes use the OpenGL libraries developed by Sun and currently supported by Microsoft.

Visualize Features: The Visualize features can be used to: 1) make any series, group, or individual riser transparent, 2) suppress any series, group or individual riser (with optional rescaling), or 3) apply a glow effect to any series, group, or individual riser.

Curve Fits: PGSDK supports Mean, Median and Standard Deviation curve fits. Advanced curve fitting: Nth-order polynomials, Best-Fit, Error-bars, etc.

Multi-Dimensional Labels: PGSDK supports multi-dimensional (also known as “hierichal”) ordinal axis labels.

User Interface Dialogs: A complete set of Microsoft Office-style Tabbed Dialogs are built directly into PGSDK for use in custom applications. It includes a Chart Type dialog with a customizable gallery of chart templates. It also includes Grids & Scales, Titles & Labels, Chart Options, Series Options, and a complete Formatting dialog with built-in gradients, textures, and pictures!

Import/Export Chart from/to JAVA Script Files: Based on special customer request, PGSDK includes API functions that can be used to read and write Java Script Files that are compatible with the Three |D| Graphics Perspective for Java product. For an export operation, you can write the chart "look" only or the graph and its data.

What you get...

The contents of the PGSDK will be slightly different depending on your development platform. It will always include:

- One or more dynamic link library (.DLL) files. These files must be included with your application's distribution package.
- One or more library (.LIB) files. These files provides access to the PGSDK API calls.

- One or more header (.H) files. These files contain all PGSDK declarations, function prototypes, data structures, and constants to compile your application.

- Sample Files: Numerous programming examples show how to access and use API calls and implement features in your program. All of the *Features* described above are demonstrated in the sample programs. Each program is fully commented. Examples are included for C, C++, Visual Basic, Power Builder, and Actor.

- Documentation: The following documents describe the Presentation Graphics SDK:

  1. *Presentation Graphics SDK Programmer's Manual* (this manual): Describes now to install and use the SDK.

  2. *Presentation Graphics SDK Attributes Guide*: Describes each of the attributes that can be applied to objects in a chart.

  3. *Presentation Graphics SDK API Guide*: Describes each of the API functions that are available in the graphics library.

  These documents are provided in Hypertext Markup Language (.HTML) format. They are also available in Microsoft Word 97 (.DOC) format.

Creating a Chart

The process of creating a graph requires these steps:

1. **Initialize the Library**. Your application must initialize the library. This is only done once in your application.

```
// Status of function calls
INT16 nStatus;
// Initialize the library instance
nStatus = InitTDGLIB();
```

2. **Allocate a Graph Pointer** to set aside the memory needed for the graph.

```
// Pointer to a graph object
GraphPtr gpGraph;
gpGraph = AllocGraphPtr();
```

3. **Allocate a Draw Environment (DE)** to set aside memory needed to describe device context (DC) information.

```
//Pointer to standard draw environment
DrawEnvPtr gpDrawEnv;
gpDrawEnv = AllocDrawEnvPtr(hwnd, DE_PORT_NORMAL);
```

4. **Connect a Data Source.** The library calls functions in your application to retrieve the data for charting and the strings used for titles and labels. This example uses the data callback method of accessing data.

```
BOOL PUBLIC InitDataCallBacks(HINSTANCE hInst,
GraphPtr pGraph, INT16 Rows, INT16 Cols)
{
     GraphDataInfo graphData;
     GetGraphDataInfo(pGraph, &graphData);
     if (!graphData.lpfnGetData)
     {
          graphData.lpfnGetString =
          (LPFNGetStringCallBack)GetStringCallback;
          graphData.lpfnGetLabel =
          (LPFNGetLabelCallBack)GetLabelCallback;
          graphData.lpfnGetData =
          (LPFNGetDataCallBack)GetDataCallback;
          graphData.lpfnSetPage = NULL;
          graphData.lpfnRealToString = NULL;
     }
     graphData.nPages = 1;
     graphData.nRows = 4; // # of Series
     graphData.nCols = 3; // # of Groups
     graphData.lClientData = (UINT32) pGraph;
     graphData.nDataIFMode = DATAIF_CALLBACK;
     return SetGraphDataInfo(pGraph, &graphData);
}
```

The data callback version is just one of the data interfaces available with the graphics library. It is used here as an example for setting the data callbacks in an application.

5. **Load a Chart Template File** into the graph structure. Loading the template file initializes the graph pointer structure with information describing the chart appearance.

```
INT16 Load_TIFFGraph (
     GraphPtr pGraph,/* Pointer to a valid graph */
     UINT32 ulRequest,
     #if defined(WIN32) || UNIX
          // Pointer to "C" style string file name
          char * pszFileName
     #else
          // Pointer to MAC File Specification
          FSSpec * pFileSpec
     #endif
);
```

If you provide the ability to select and modify the attributes of chart objects, a selection list pointer must be initialized.

```
 SelListPtr gpList; /* ptr to Selection List */
/* allocate space for the selection list */
gpList = Select_AllocList();
```

6. **Set the Destination Rectangle.** The destination rectangle defines (in pixels) the coordinates on the output device context to image the chart.

```
// Rectangle in the client area
Rect rClient;
rClient.top=0;
rClient.left=0;
rClient.bottom=480;
rClient.right=640;
// Pointer to a draw environment
DrawEnvPtr gpDrawEnv;
// Set rectangle
DESetDestRect(gpDrawEnv, &rClient );
```

7. **Draw the graph.** Drawing the chart is normally accomplished as a result of a WM_PAINT message from Windows.

```
void CMysampView::OnDraw(CDC* pDC)
{
   CMysampDoc* pDoc = GetDocument();
   ASSERT_VALID(pDoc);
   Rect r;
   GetClientRect(m_Bounds);
   SetRect(&r, m_Bounds.left,m_Bounds.top,
        m_Bounds.right,m_Bounds.bottom);
   if (m_GraphValid && !m_Bounds.IsRectEmpty())
   {
        /* SET CHART SIZE & LOCATION IN WINDOW */
        DESetDestRect(m_pDrawEnv, &r);
        /* SET DRAWING MODE TO NORMAL,ON-SCREEN */
        DESetPortInfo(m_pDrawEnv,(PORTWINDOW)
             pDC->GetSafeHdc(),DE_PORT_NORMAL);
        DrawBegin(m_pDrawEnv); /* DRAW THE GRAPH */
        DrawTheGraph(m_pDrawEnv, pDoc->m_pGraph,
             TRUE, TRUE, TRUE);
        DrawEnd(m_pDrawEnv);
   }
}
```

User Selections

If you provide the ability to select and modify the attributes of chart objects, a selection list pointer must be initialized. Example:

```
CComSelList::CComSelList()
{
     m_pSelList = ::Select_AllocList();
     m_pDetNodeRef= ::AllocPtrClear(sizeof(DetNodeRef ));
     m_nDragHandle = HANDLE_NONE;
}
```

User selections are made by the detection of a mouse click on a chart object. During the chart drawing process a detection node (*detnode*) is created for each object on the screen. A detnode describes the object, e.g. its position on the screen, series and group number, color, font, line size, object type, etc. Detnodes are created for all the objects on the chart, title, frame, risers, legend, grid lines, areas, etc.

The detnode is passed to a selection list API to add the object to the selection list. A selection list is a linked list of user-selected objects. This allows the user to use SHIFT-left-click to keep adding objects to the selection list and modify an attribute to effect all selected objects. This sample adds and removes items from a selection list:

```
void CComSelList::SelectObject( CComDrawEnv& de,
     BOOL bShiftKey, CComPoint pt )
{
     SelItemPtr pSelItem;
     // change to virtual coordinate
     de.DeviceToVirtual( pt );
     if (::FindDetNode ( (GraphPtr) m_pGraph, &pt,
           (DetNodeRefPtr) m_pDetNodeRef))
     {
          // if shift pressed, search to see if
          // already selected. If it is, deselect.
          // If not, add it to the selection list.
          // -------------------------------------
          if (bShiftKey)
          {
               pSelItem = ::Select_SearchList (
               (SelListPtr) m_pSelList,
               (DetNodeRefPtr) m_pDetNodeRef);
               if (pSelItem)
               {
                    XOR_SelItem( de, pSelItem );
                    ::Select_RemoveItem (
                    (SelListPtr) m_pSelList,
                    (SelItemPtr) pSelItem);
               }
               else AddItemToList( de, FALSE );
          }

          /* NO SHIFT KEY */
          else
          {
               pSelItem = ::Select_SearchList(
               (SelListPtr) m_pSelList,
               (DetNodeRefPtr) m_pDetNodeRef);
               if(!pSelItem)AddItemToList(de,TRUE);
          }
     }
}
```

```
//========================================
// CComSelList::AddItemToList
//========================================
void CComSelList::AddItemToList(
     CComDrawEnv& de, BOOL bExclusive )
{
     SelItemPtr pSelItem;
     XOR_SelList( de );
     if( bExclusive ) ::Select_ClearList(
          (SelListPtr) m_pSelList );
          pSelItem = ::Select_AddItem (
          (GraphPtr) m_pGraph,
          (SelListPtr) m_pSelList,
          (DetNodeRefPtr) m_pDetNodeRef );

     // a curve has more than one detnode (with the
     // same objectID) add all related objects
     // ----------------------------------
     if( pSelItem->nObjectID == O2D_CURVE )
     {
          ::Select_AddRelatedItems((GraphPtr) m_pGraph,
          (SelListPtr) m_pSelList,
          (DetNodeRefPtr) m_pDetNodeRef,
          SLR_SAME_OBJECT,FALSE,O2D_CURVE );
     }

     ::Select_CalcHandlesItem((DrawEnvPtr)de.GetDrawEnv(),
          (SelItemPtr) pSelItem );
     XOR_SelList( de );}
```

Highlighting objects in the selection list is also easy. Example:

```
void CComSelList::XOR_SelItem(
     CComDrawEnv& de, void * pSelItem )
{
     ::Select_DrawXorItem((DrawEnvPtr)de.GetDrawEnv(),
          (SelItemPtr) pSelItem );
}
```

Modifying Attributes

Modifying object attributes can change the appearance of the chart. The library defines over 200 attributes that specify the appearance of objects in a chart. In functionality, the attributes range from draw/no draw attributes to size/location attributes to color and special effects. There are basically six API functions that provide the interface to get and set chart object attributes.

These API functions can be used to determine the current or default value of an attribute:

- GetGraphAttr(); Get attribute information for a specific graph object
- GetGraphAttrSI(); Get attribute information for a specific item in a selection list

- GetGraphAttrSL(); Get attribute information for the first item in a selection list

The GetGraphAttr() function lets you request attribute information for a specific object. The following example requests the color (A_FONTCOLOR_RGB) attribute of the graph's title object (O5D_LBLTITLE):

```
nRet = GetGraphAttr(pGraph, GRAPH_LAYER,
     O5D_LBLTITLE,NULL_SERIESID, NULL_GROUPID,
     A_FONTCOLOR_RGB,(void FAR *)&titlecolor);
```

Rather than identifying a specific Object ID, the GetGraphAttrSI() function lets you choose the object as an item in a selection list. The following example requests the color of the font (A_FONTCOLOR_RGB) for the item in the selection list at *gpSelItem*:

```
nRet = GetGraphAttrSI (
     gpGraph,         /* graph pointer */
     gpSelItem,       /* pointer to selection item */
     A_FONTCOLOR_RGB, /* attribute to modify */
     &rgbOld          /* old attribute value */
);
```

The GetGraphAttrSL() function requests attribute information for the first item in a selection list. The following example requests the color of the font (A_FONTCOLOR_RGB) for the first item in the selection list at *gpList*:

```
nRet = GetGraphAttrSL(
     gpGraph,          /* graph pointer */
     gpList,           /* pointer to selection list */
     A_FONTCOLOR_RGB,  /* attribute to modify */
     &rgbOld           /* old attribute value */
);
```

There are three API functions that can be used to set the value of an attribute:

- SetGraphAttr(); Set attribute information for a specific object

- SetGraphAttrSI(); Set attribute information for a specific item in a selection list

- SetGraphAttrSL(); Set attribute information for all items in a selection list where the attribute is applicable.

In all three cases, you identify an attribute by its attribute code and specify attribute information that you want to apply to an object in the graph. The SetGraphAttrSL function will only choose items in the selection list that are appropriate to the attribute being used. For example, the A_AREACOLOR_RGB attribute can only be applied to area objects. If there are no area objects in the selection list, the SetGraphAttrSL() function is ignored and a failure status is returned. Also note that other factors/attributes may eliminate some items from being chosen by SetGraphAttrSL(). If an item's A_AREACOLOR_NONE attribute is applied (for example), the color cannot be changed by A_AREACOLOR_RGB.

The following example sets the color attribute (A_FONTCOLOR_RGB) of the graph's title object (O5D_LBLTITLE) to the color values stored at *titlecolor*.

```
//set the color of the title to red
nRet = SetGraphAttr(pGraph, GRAPH_LAYER,
     O5D_LBLTITLE, NULL_SERIESID, NULL_GROUPID,
     A_FONTCOLOR_RGB, (void FAR *)&titlecolor
);
```

The following example sets the alignment (A_FONTALIGN) of a text object at *gpSelItem* to right justified (*&Right*):

```
nRet=SetGraphAttrSI(
     gpGraph,gpSelItem, A_FONTALIGN,&Right
);
```

The following example sets the font color (A_FONTCOLOR_RGB) for all selected text items in the selection list at *gpList* to the color values stored at *aRGB[i]*:

```
nRet = SetGraphAttrSL (
     gpGraph, gpList, A_FONTCOLOR_RGB, &aRGB[i]
);
```

Chart Attributes

- Common Attributes used in all charts
- 2D Attributes
- 3D Attributes
- Advanced (OpenGL) Attributes
- Annotation Attributes
- ABA_ Bar/Area Attributes

- Color Model Attributes
- Error Bar Attributes
- Multi-Dimensional Label Attributes
- Pie Chart Attributes
- Polar Chart Attributes
- Series/Group Attributes
- Stock Chart Attributes
- Table Chart Attributes

Common Attributes

Common attributes can be applied to any of the chart types.

An area on a chart can be a riser, marker, wall, or background. Any part of the chart that has a surface is considered an area

A_AREABACKCOLOR; Area background color

A_AREACOLOR_ALPHA; Opacity/Transparency for areas

A_AREACOLOR_NONE; Area Color None (transparent)

A_AREACOLOR_RGB; Area Color definition

A_AREADROPSHADOW; Area Drop Shadow

A_AREAPATTERN; Area pattern definition

A_AREASFX; Area Special Effects definition

Font attributes can be applied to any text on the chart:

A_FONT_FANCYBOX; Font Fancy Box definition

A_FONT_LEADING; Font Leading

A_FONTALIGN; Font Alignment

A_FONTBACKCOLOR; Font Background Color

A_FONTCHARSET; Font Character Set

A_FONTCOLOR_NONE; Font Color None (transparent)

A_FONTCOLOR_RGB; Font Color definition

A_FONTDROPSHADOW; Font Drop Shadow

A_FONTNAME; Font Name

A_FONTNAME_MBC; Font Name using mix single-byte fonts

A_FONTORIENT; Font Orientation

A_FONTORIENTCUSTOM; Font Orientation Custom angle

A_FONTPATTERN; Font Pattern definition

A_FONTSFX; Font Special Effects definition

A_FONTSIZE_LIMIT; Font Size Limit

A_FONTSIZE_POINT_100; Font Point Size in 100th/Point Units

A_FONTSIZE_VC; Font Size in Virtual Coordinates

A_FONTSTYLE; Font Style (underline, bold, etc.)

Glow attributes apply an alpha-channel color/glow to riser objects.

A_GLOW; Glow data objects mode on/off

A_GLOW_COLOR; Glow data object's color

A_GLOW_MODE; Glow inside, outside, both

A_GLOW_THICKNESS; Glow thickness

This attribute defines the graph type (3D, bar, line, area, pie, etc.) and sub-type:

A_GRAPH_PRESET; Graph Preset (graph type and subtype)

Line attributes can be applied to any line on the chart

A_LINECOLOR_ALPHA; Opacity/Transparency for lines

A_LINECOLOR_NONE; Line Color None (transparent)

A_LINECOLOR_RGB; Line Color definition

A_LINEPATTERN; Line Pattern (e.g. solid, dash, dot, dot and dash)

A_LINESFX; Line Special Effects definition

A_LINEWIDTH; Line Width (in virtual coordinates)

A_LINEWIDTH_POINT_100; Line Width in 100th/Point Units

Location attributes are available for the title, subtitle, and footnote:

A_LOCATE_FOOTNOTE; Locate Footenote (position/size of bounding box)

A_LOCATE_FRAME; Locate Frame (position/size of chart frame)

A_LOCATE_SUBTITLE; Locate Subtitle (position/size of bounding box)

A_LOCATE_TITLE; Locate Title (position/size of bounding box)

Other common attributes include:

A_LOCK_FONTSIZE; Lock Font Size

A_SHOW_FOOTNOTE; Show/Hide Footnote

A_SHOW_SUBTITLE; Show/Hide Subtitle

A_SHOW_TITLE; Show/Hide Title

A_SUPPRESS; Suppress data

2D Attributes

2-D attributes are those that relate specifically to two-dimensional graph types, bar, line, area, and scatter (XY). Most of the 2-D attributes control the appearance of the chart as a whole:

ATTRIBUTE	**Description**

A2D_ADVANCE_CURVEFITS	Advanced Curvefits on a particular series
A2D_AUTOFIT_ALL	Autofit All (global autofitting)
A2D_AUTOFIT_DATATEXT	Autofit Data Text
A2D_AUTOFIT_LEGEND	Autofit Legend
A2D_AUTOFIT_O1/_O2	Autofit labels on an ordinal axis
A2D_AUTOFIT_X/_Y1/_Y2	Autofit labels on a numeric axis
A2D_BAR_GROUP_SPACING	Bar Group Spacing in side by side bar chart
A2D_BAR_RISER_WIDTH	Bar Riser Width of all risers in a bar chart
A2D_BASEVALUE_X/_Y1/_Y2/_Y3/_Y4	Define the base value from which values are drawn on a numeric axis
A2D_BUBBLEGRID_COUNT_X	Bubble Grid Count on the X-axis
A2D_BUBBLEGRID_COUNT_Y	Bubble Grid Count on the Y-axis
A2D_BUBBLEGRID_POSITION_X	Bubble Grid Position on the X-Axis
A2D_BUBBLEGRID_POSITION_Y	Bubble Grid Position on the Y-Axis
A2D_COLORBYSERIES	Color By Series or group
A2D_CONNECT_STACKBARS	Connect bars in a stacked bar chart
A2D_CURVECOLORASSERIES	Curve Color As Series
A2D_CURVEMOVING	Curve Moving average line data points
A2D_CURVESMOOTH	Curve Smooth data points
A2D_DATAFORMAT	Data Format of data input to a 2D graph
A2D_DATAFORMAT_VERT	Data Format Vertical orientation of data points
A2D_DEPTH_IMS_ANGLE	Depth Angle of 3D effect on surface areas of a 2D chart
A2D_DEPTH_IMS_THICK	Depth Thickness of 3D effect on a 2D chart
A2D_DEPTH_MODE	Depth Mode of 3D effect on a 2D chart
A2D_DIRECTION_X/_Y1/_Y2/_Y3/_Y4/_Z	Direction and order of a numeric axis
A2D_EXCLUDE_ZERO_X/_Y1/_Y2/_Y3/_Y4/_Z	Exclude zero as numeric axis origin based on data range
A2D_FORMAT_SDLINE	Format Series-Dependent Line formula and coefficient values
A2D_FORMAT_SDLINE_ADV	Format Series-Dependent Line formula and coefficient values advanced
A2D_FORMAT_X/_Y1/_Y2/_Y3/_Y4/_Z	Format numeric axis labels
A2D_FORMAT_X/_Y1/_Y2/_Y3/_Y4/_Z_ADV	Advanced formatting of numeric axis labels
A2D_FORMATDTXT_X/_Y1/_Y2/_Y3/_Y4	Format Data Text on a numeric axis
A2D_FORMATDTXT_X/_Y1/_Y2/_Y3/_Y4_ADV	Advanced formatting of data text on a numeric axis
A2D_GRID_STEP_X/_Y1/_Y2/_Y3/_Y4/_Z	Grid Steps on a numeric axis
A2D_GRIDLINES_ON_TOP	Gridlines On Top of all other objects in the chart
A2D_GRIDLINES_X/_Y1/_Y2/_Y3/_Y4/_Z	Gridlines on a numeric axis
A2D_GRIDLINESORD_O1/_O2	Gridlines on an ordinal axis
A2D_GRIDMODEMAJOR_O1	Grid Mode Major on the O1-Axis
A2D_GRIDMODEMINOR_O1	Grid Mode Minor on the O1-Axis
A2D_HL_COLOR	Highlight Color of a group in a series
A2D_IGNORE_GROUPS	Ignore Group

A2D_IGNORE_SERIES	Ignores Series
A2D_LABELMODE_O1/O2	Label Mode on an ordinal axis
A2D_LABELWRAPLINES_O1	Label Wrap Lines on the O1-axis
A2D_LABELWRAPMODE_O1	Label Wrap Mode on the O1-axis
A2D_LEGEND_FANCYBOX	Legend Fancy Box definition
A2D_LEGEND_LOCK	Legend Placement
A2D_LEGEND_MARKER_ORIENTATION	Legend Marker Orientation
A2D_LOCATE_LEGEND	Locate Legend
A2D_LOCATE_LINR_TEXT	Locate Linear Regression Text
A2D_LOCATE_TITLE_X/_Y1/_Y2/_Y3/_Y4	Locate a numeric axis title
A2D_LOG_X/_Y1/_Y2/_Y3/_Y4	Logarithmic/linear scaling on a numeric axis
A2D_MANUAL_PICTOGRAPH_SCALE_VALUE	Unit/Scale to stack pictograms
A2D_MARKERSHAPE	Marker Shape in a 2D Chart
A2D_MARKERSHAPE_SD	Marker Shape in a 2D Chart for a series
A2D_MARKERSIZE	Marker Size in a 2D Chart
A2D_MARKERSIZE_SD	Marker Size in a 2D Chart for a series
A2D_MOVE_SERIES	Move series in a stacked bar chart
A2D_MULTI_Y_INFO	Define Multi-Y axis information
A2D_ORD_SPACE_MODE	Ordinal Axis Space Mode
A2D_ORIENTATION	Orientation (horizontal/vertical) of chart
A2D_PLACE_O1_/O2	Place data value on risers on an ordinal axis
A2D_PLACE_X/_Y1/_Y2/_Y3/_Y4	Place data values on risers on a numeric axis
A2D_PRESDL_CONN	Pre-Series Dependent connection line
A2D_PRESDL_CONNBRK	Pre-Series Dependent connection line with breaks for missing data points
A2D_PRESDL_STEP	Pre-Series Dependent step connection line
A2D_PRESDL_STEPBRK	Pre-Series Dependent step connection line with breaks for missing data points
A2D_PRESDL_STEPVERT	Pre-Series Dependent vertical step connection line
A2D_SCALE_X/_Y1/_Y2/_Y3/_Y4/_Z	Scaling on a numeric axis
A2D_SCALEBASE	Scale Base of negative risers in a 2D chart
A2D_SCALEEND_O1/_O2	Scale End on an ordinal axis
A2D_SCALEEND_X/_Y1/_Y2/_Y3/_Y4/_Z	Scale End on a numeric axis
A2D_SCALEFREQ_BEG_O1/_O2	Scale Frequency Begin on an ordinal axis
A2D_SCALEFREQ_BEG_X/_Y1/_Y2/_Y3/_Y4	Scale Frequency Begin on a numeric axis
A2D_SCALEFREQ_END_O1/_O2	Scale Frequency End on an ordinal axis
A2D_SCALEFREQ_END_X/_Y1/_Y2/_Y3/_Y4	Scale Frequency End on a numeric axis
A2D_SCALEFREQ_O1/_O2	Scale Frequency of labels on an ordinal axis
A2D_SCALEFREQ_X/_Y1/_Y2/_Y3/_Y4	Scale Frequency of labels on a numeric axis
A2D_SCIMOVAVG	Scientific Moving Average curve definition
A2D_SD_SHOWDATATEXT	Series-Dependent Show/Hide DataText

A2D_SDDATALINE_TYPE	Series-Dependent Data Line Type (markers, lines, or both)
A2D_SDEMPHASIZED	Series-Dependent Emphasize a series
A2D_SDLINECONN	Series-Dependent Line Connecting a series
A2D_SDLINECURV	Series-Dependent Line Curve connecting a series
A2D_SDLINELINR_CORR	Series-Dependent Correction coefficient text
A2D_SDLINELINR_EXP	Series-Dependent Exponential linear regression line
A2D_SDLINELINR_FORMULA	Series-Dependent Formula text for the linear regression lines
A2D_SDLINELINR_LINE	Series-Dependent Linear Regression Line
A2D_SDLINELINR_LOG	Series-Dependent common log linear regression line
A2D_SDLINELINR_NATLOG	Series-Dependent Natural Log linear regression line
A2D_SDLINELINR_NPOLY	Series-Dependent Polynomial fit line
A2D_SDLINELINR_NPOLYFAC	Series-Dependent degree of linear regression for the polynomial fit line
A2D_SDLINEMEAN	Series-Dependent Mean average line
A2D_SDLINEMOVA	Series-Dependent Moving Average line
A2D_SDLINESTDD	Series-Dependent Standard Deviation line
A2D_SERIES_LABELS_ON_CHART	Series Labels to the right or left of Chart frame
A2D_SHOW_BUBBLEGRID	Show/Hide bubble grid
A2D_SHOW_DATATEXT	Show/Hide Data Text
A2D_SHOW_DATATEXT_CONNECTLINE	Show/Hide connecting line between data text and riser/marker
A2D_SHOW_DIVBIPOLAR	Show/Hide line to divide a bipolar chart
A2D_SHOW_ERRORBAR	Show/Hide Error Bars
A2D_SHOW_LEGEND	Show/Hide Legend in a 2D chart
A2D_SHOW_O1_/O2	Show/Hide ordinal axis line and text headers
A2D_SHOW_OFFSCALE_X/_Y1/_Y2/_Y3/_Y4	Show/Hide Off-Scale values on a numeric axis
A2D_SHOW_QUADRANTS	Show/Hide Quadrants in a Bubble Chart
A2D_SHOW_TITLE_SERIES	Show/Hide Series Title
A2D_SHOW_TITLE_X/_Y1/_Y2/_Y3/_Y4	Show/Hide a numeric axis title
A2D_SHOW_X/_Y1/_Y2/_Y3/_Y4	Show/Hide a numeric axis
A2D_SHOW_ZL_X/_Y1/_Y2/_Y3/_Y4	Show/Hide zero line on a numeric axis
A2D_SIDE_O1/_O2	Side of chart ordinal axis labels are displayed
A2D_SIDE_X/_Y1/_Y2/_Y3/_Y4	Side of chart numeric axis labels are displayed
A2D_SIZE_ERRORBARS	Size of Error Bars
A2D_SPLITY_ND	Assign series to Y1 or Y2 axis in a BiPolar Chart
A2D_SPLITY_POSITION	Split Y1/Y2 position in a BiPolar Chart
A2D_SQUARE_LGND_ICONS	Square Legend Icons
A2D_STAGGER_O1/_O2	Stagger ordinal axis labels
A2D_STAGGER_X/_Y1/_Y2/_Y3/_Y4/_Z	Stagger numeric axis labels
A2D_SYMBOL_FORSERIES	Symbol For Series in a 2D chart

A2D_UNIFORM_QDRBORDERS	Uniform Quantitative Data Representation (Riser) Border colors
A2D_UNIFORM_QDRSHAPES	Uniform Quantitative Data Representation (Riser) Shapes
A2D_USE_MANUAL_PICTOGRAPH_SCALE	Enable/Disable manual pictograph scaling
A2D_USERLINE_AXIS	User-Specified girdline on an axis
A2D_USERLINE_SHOW	User-Specified gridline show/hide
A2D_USERLINE_VALUE	User-Specified gridline location

3D Attributes

3D attributes relate to the 3D graph types. Most of them control the appearance of the chart as a whole:

ATTRIBUTE	Description
A3D_AUTOFIT	Autofit entire cube when 3D chart is rotated
A3D_AUTOFIT_DATALABELS	Autofit Data Labels in a 3D chart
A3D_AUTOFIT_LEGEND	Autofit Legend in a 3D chart
A3D_AUTOFIT_TEXT	Autofit Text in a 3D chart
A3D_AUTOSHADE_CUBE	Auto Shade the 3D cube in a 3D chart
A3D_AUTOSHADE_RISERS	Auto Shade the 3D risers in a 3D chart
A3D_AXIS_DIVS_X	Axis Divisions-X on cube walls in a 3D chart
A3D_AXIS_DIVS_Y	Axis Divisions-Y on cube walls in a 3D chart
A3D_AXIS_DIVS_Z	Axis Divisions-Z on cube walls in a 3D chart
A3D_CUSTOMVIEW	Custom View angle in a 3D chart
A3D_DATAFORMAT	Data Format of data input to a 3D graph
A3D_FIXED_2D_TEXT	Fixed 2D Text or 3D appearance sizing in a 3D chart
A3D_FONTBOX	Font Box definition in a 3D chart
A3D_FONTMARGIN	Font Margin between text object and its box
A3D_FONTOFFSET	Font Offset from axis to text object
A3D_FONTSIZE	Font Size of a text object in a 3D chart
A3D_FONTWIDTH	Font Width of a 3D text object relative to its height
A3D_FORMAT_X	Format X-axes labels in a 3D chart
A3D_FORMAT_Y	Format Y-axes labels in a 3D chart
A3D_FORMAT_Z	Format Z-axes labels in a 3D chart
A3D_GLOW_RISER	Glow Risers in a 3D chart
A3D_GRAPH_BY_WHAT	Graph By What (series, group, or series & group)
A3D_GRID_FLOOR	Show/Hide gridlines on 3D cube floor
A3D_GRID_LWALL	Show/Hide gridlines on 3D cube left wall
A3D_GRID_RISER	Show/Hide gridlines on 3D risers
A3D_GRID_RWALL	Show/Hide gridlines on 3D cube right wall
A3D_IGNORE_SERIES	Ignore Series in a 3D chart
A3D_INSET_LGND_ICON	Inset Legend Icon in a 3D chart

A3D_INSET_LGND_TEXT	Inset Legend Text in a 3D chart
A3D_JUSTIFY	Justifiy 3D text object
A3D_LOCATE_COLTITLE_2D	Locate 2D column title in a 3D chart
A3D_LOCATE_LEGEND	Locate Legend in a 3D chart
A3D_LOCATE_LYTITLE_2D	Locate 2D left Y1-axis title in a 3D chart
A3D_LOCATE_ROWTITLE_2D	Locate 2D row title in a 3D chart
A3D_LOCATE_RYTITLE_2D	Locate 2D right Y1-axis title in a 3D chart
A3D_MARKER_SHAPE	Marker Shape in a 3D chart
A3D_MARKER_SHAPE_DEFAULT	Marker Shape Default in a 3D chart
A3D_MARKER_SIZE	Marker Size in a 3D chart
A3D_RESCALE_ON_SUPPRESS	Rescale On Suppress after visualize attributes
A3D_RISERTHICK	Riser Thickness of risers in a 3D chart
A3D_SCALEBASE	Scale Base of netative risers in a 3D chart
A3D_SCALEEND_X	Scale End on the X-axis in a 3D chart
A3D_SCALEEND_Y	Scale End on the Y-axis in a 3D chart
A3D_SCALEEND_Z	Scale End on the Z-axis in a 3D chart
A3D_SCALEFREQ_X	Scale Frequency on the X-axis in a 3D chart
A3D_SCALEFREQ_Y	Scale Frequency on the Y-axis in a 3D chart
A3D_SCALEFREQ_Z	Scale Frequency on the Z-axis in a 3D chart
A3D_SCALERANGE_X	Scale Range on the X-axis in a 3D chart
A3D_SCALERANGE_Y	Scale Range on the Y-axis in a 3D chart
A3D_SCALERANGE_Z	Scale Range on the Z-axis in a 3D chart
A3D_SCALETYPE_X	Scale Type (logrithmic or linear) on the X-axis in a 3D chart
A3D_SCALETYPE_Y	Scale Type (logrithmic or linear) on the Y-axis in a 3D chart
A3D_SCALETYPE_Z	Scale Type (logrithmic or linear) on the Z-axis in a 3D chart
A3D_SCATTER_N	Scatter line connecting markers in a series
A3D_SCATTER_X	Scatter line from each marker to left wall of a 3D cube
A3D_SCATTER_Y	Scatter line from each marker to floor of a 3D cube
A3D_SCATTER_Z	Scatter line from each marker to right wall of a 3D cube
A3D_SERIES_TYPE	Series Type selects riser type for a series or group in a 3D chart
A3D_SHOW_FLOOR	Show/Hide Floor of the 3D cube
A3D_SHOW_LEGEND	Show/Hide Legend in a 3D chart
A3D_SHOW_LWALL	Show/Hide Left Wall of the 3D cube
A3D_SHOW_RWALL	Show/Hide Right Wall of the 3D cube
A3D_SHOWBOX_DATALABELS	Show/Hide box around data labels in a 3D chart
A3D_SHOWOFFSCALE_X	Show/Hide offscale values on the X-axis in a 3D chart
A3D_SHOWOFFSCALE_Y	Show/Hide offscale values on the Y-axis in a 3D chart

A3D_SHOWOFFSCALE_Z	Show/Hide offscale values on the Z-axis in a 3D chart
A3D_SHOWTEXT_ALL	Show/Hide all text in a 3D chart
A3D_SHOWTEXT_COLHEADERS	Show/Hide Column Headers in a 3D Chart
A3D_SHOWTEXT_COLTITLE	Show/Hide Column Title in a 3D Chart
A3D_SHOWTEXT_DATALABELS	Show/Hide Data Labels in a 3D Chart
A3D_SHOWTEXT_LEFTNUMBERS	Show/Hide labels next to left wall of 3D cube
A3D_SHOWTEXT_LEFTTITLE	Show/Hide title next to left wall of 3D cube
A3D_SHOWTEXT_RIGHTNUMBERS	Show/Hide labels next to right wall of 3D cube
A3D_SHOWTEXT_RIGHTTITLE	Show/Hide title next to right wall of 3D cube
A3D_SHOWTEXT_ROWHEADERS	Show/Hide row headers in a 3D chart
A3D_SHOWTEXT_ROWTITLE	Show/Hide Row Title in a 3D chart
A3D_SQUARE_LGND_ICONS	Square Legend Icons in a 3D chart
A3D_SUPPRESS_RISER	Suppress Riser(s) in a 3D chart
A3D_SYMBOL_FORSERIES	Symbol For Series in a 3D chart
A3D_TEXT_POSITION	Text Position in a 3D chart
A3D_TEXT_QUALITY	Text Quality of headers, titles, & labels in a 3D chart
A3D_TRANS_RISER	Transparent Riser(s) in a 3D chart
A3D_UNIFORM_QDRSHAPES	Uniform Quantative Data Representation (Riser) Shapes
A3D_VIEWANGLES	Viewing Angles of the 3D cube
A3D_VIEWSTRUCT	Viewing Structure definition

Advanced Drawing Attributes

ATTRIBUTE	Description
A3D_ADV_2DDEPTH_DATA	Advanced 2D Depth of the data layer
A3D_ADV_2DDEPTH_FRAME	Advanced 2D Depth of Frame
A3D_ADV_2DDEPTH_LEGEND	Advanced 2D Depth of Legend
A3D_ADV_2DFRAMEFOOT	Advanced 2D Frame Foot definition
A3D_ADV_2DFRAMEMARGIN	Advanced 2D Frame Margin definition
A3D_ADV_2DVIEWER_DIST	Advanced 2D Viewer Distance
A3D_ADV_2DVIEWER_X	Advanced 2D Viewer X
A3D_ADV_2DVIEWER_Y	Advanced 2D Viewer Y
A3D_ADV_ANTIALIAS	Advanced Antialiasizing
A3D_ADV_BGRADIOSITY	Advanced Background Radiosity (back lighting)
A3D_ADV_CHARTLIGHT_1	Advanced Chart Light 1 definition
A3D_ADV_CHARTLIGHT_2	Advanced Chart Light 2 definition
A3D_ADV_CHARTLIGHT_3	Advanced Chart Light 3 definition
A3D_ADV_DROP_SHADOW	Advanced Drop Shadow around a Pie Chart
A3D_ADV_DEFAULT_MATERIAL	Advanced Default Material for objects
A3D_ADV_DOADV_CHART	Advanced Do Advanced Drawing of Chart
A3D_ADV_DOADV_TITLES	Advanced Do Advanced Drawing of Chart Titles
A3D_ADV_FLOATER_VARIANT	Advanced risers types in a 3D floating cube chart

ATTRIBUTE	Description
A3D_ADV_FOV	Advanced Field-of-View (FOV)
A3D_ADV_HIFI_OBJECTS	Advanced HiFi Objects
A3D_ADV_LABEL_MATERIAL	Advanced Label Material
A3D_ADV_LIGHTAMBIENT	Advanced Light Ambient
A3D_ADV_LOFT_SHAPE	Advanced Loft Shape of risers
A3D_ADV_LOFT_TWIST	Advanced Loft Twist of risers
A3D_ADV_MATAMBIENT	Advanced Material Ambient light reflecting from default material
A3D_ADV_MATGLOW	Advanced Material Glow from the default material
A3D_ADV_PIE_EDGETYPE	Advanced Pie Edge Style
A3D_ADV_PIE_FEELERTYPE	Advanced feeler type in a pie chart
A3D_ADV_PIE_LABELSFACE	Advanced Pie Labels Face (How much pie labels are turned to face the camera)
A3D_ADV_PIE_TINTLABELS	Advanced Pie Tint Labels
A3D_ADV_RISER_SIDES	Advanced Riser Sides
A3D_ADV_ROUNDING	Advanced Rounding to apply to advanced 3D generated objects
A3D_ADV_TEXT_STYLE	Advanced Text Styles
A3D_ADV_TITLELIGHT	Advanced Title Light definition

Annotation Attributes

ATTRIBUTE	Description
AAN_ARC_INFO	Annotation Arc Information
AAN_ARW_ANGLE	Annotation Arrow Angle
AAN_ARW_HDIND	Annotation Arrow Head Indent
AAN_ARW_HDWID	Annotation Arrow Head Width
AAN_ARW_PRESET	Annotation Arrow Preset
AAN_ARW_SIDES	Annotation Arrow Sides
AAN_ARW_TLIND	Annotation Arrow Tail Indent
AAN_ARW_TLWID	Annotation Arrow Tail Width
AAN_BOUNDS	Annotation Bounds
AAN_BOXTYPE_BALLOON	Annotation Box Type Balloon
AAN_BOXTYPE_BCHART	Annotation Box Type Bullet Chart Text
AAN_BOXTYPE_BOX	Annotation Box Type Box
AAN_BOXTYPE_DBLTBOX	Annotation Box Type Double Edged Text Box
AAN_BOXTYPE_NOBOX	Annotation Box Type No Box
AAN_BOXTYPE_THREEDTBOX	Annotation Box Type 3D Text Box
AAN_BULLET_SHP	Annotation Bullet Shape
AAN_BULLET_SIZE	Annotation Bullet Size
AAN_BULLET_SPACING	Annotation Bullet Spacing
AAN_ENDPOINTS	Annotation Endpoints of arrow or line
AAN_LINEARROWHEADS	Annotation Line Arrowheads
AAN_MOVE	Annotation Move

ATTRIBUTE	Description
AAN_RECT_ROUNDNESS	Annotation Rectangle Roundness
AAN_RROUNDNESS_100	Annotation Rectangle Rounding 100%
AAN_RROUNDNESS_25	Annotation Rectangle Rounding 25%
AAN_RROUNDNESS_50	Annotation Rectangle Rounding 50%
AAN_RROUNDNESS_75	Annotation Rectangle Rounding 75%
AAN_RROUNDNESS_NONE	Annotation Rectangle Rounding None
AAN_TBOX_EDITBOUNDS	Annotation Text Box Editable Bounds
AAN_TBOX_MARGINS	Annotation Text Box Margins
AAN_TEXT	Annotation Text
AAN_THICKNESS	Annotation Thickness

Bar/Area (ABA_) Attributes

ATTRIBUTE	Description
ABA_PICTOGRAPH	Picture drawn on risers

Color Model Attributes

ATTRIBUTE	Description
ACM_AUTOMATIC_COLORMODEL	Automatic Color Model Control
ACM_COLORDIVISIONS	Color Division in Color Model
ACM_COLORMODEL	Color Model scheme
ACM_SERIES_COLOR	Series Color in a Color Model
ACM_WASH	Wash of a Color Model
ACM_WASHBEGIN	Wash Begin color of a Color Model
ACM_WASHEND	Wash End color of a Color Model

Error Bar Attributes

ATTRIBUTE	Description
AEB_ERRORBAR_MODE	Error Bar Mode
AEB_ERRORBAR_TYPE	Error Bar Type
AEB_ERRORBAR_VALUES	Error Bar Values
AEB_ERRORBAR_X_SHOWALL	Error Bars X-Axis Show All
AEB_ERRORBAR_X_SHOWSERIES	Error Bars X-Axis Show for Series
AEB_ERRORBAR_Y_SHOWALL	Error Bars Y-Axis Show All
AEB_ERRORBAR_Y_SHOWSERIES	Error Bars Y-Axis Show for Series

Multi-Dimensional Labels Attributes

ATTRIBUTE	Description
AMD_OPTIONS	Options for Multi-dimensional labels

Pie Chart Attributes

Pie attributes are those that relate specifically to the pie charts:

ATTRIBUTE	Description
API_AUTOFIT_LBLPIE	Autofit Label Pie

Pie attributes are those that relate specifically to the pie charts:

ATTRIBUTE	Description
API_DEPTH	Depth of Pie chart
API_DRAW_CLOCKWISE	Draw Clockwise Pie chart
API_FORMAT_DATATEXT	Format Data Text in a Pie Chart
API_FORMAT_DATATEXT_ADV	Format Data Text Advanced in a Pie Chart
API_FORMAT_RINGTEXT	Format Ring Text in a ring pie chart
API_FORMAT_RINGTEXT_ADV	Format Ring Text Advanced in a ring pie chart
API_HOLESIZE	Hole size of center ring in a ring pie chart
API_INSET_LBLPIE	Inset Label Pie
API_INSET_PIE	Inset Pie (margins around a pie)
API_PIESPERROW	Pies-Per-Row in a multiple pie chart
API_PLACE_DATALABEL	Place Data Label in a Pie Chart
API_PLACE_VALUELABEL	Place Value Label in a Pie Chart
API_ROTATE	Rotate Pie Chart
API_SHOW_AS_COLUMN	Show As Column a series in a pie chart
API_SHOW_FEELER	Show/Hide Feelers in a pie chart
API_SHOW_LBLFEELER	Show/Hide Label Feeler in a pie chart
API_SHOW_LBLPIE	Show/Hide Label Pie in a pie chart
API_SHOW_LBLRING	Show/Hide Label Ring in a ring pie chart
API_SLICE_DELETE	Slice Delete in a pie chart
API_SLICE_MOVE	Slice Move in a pie chart
API_SLICE_RESTORE	Slice Restore slices deleted or moved in a pie chart
API_TILT	Tilt a pie chart
API_TILT_ON	Tilt On/Off

Polar Chart Attributes

ATTRIBUTE	Description
APL_ANGLE	Angle in Degrees or Radian
APL_AXIS	Axis Show/Hide in Polar coordinate chart
APL_AXIS_CIRCLES	Axis Circles in a Polar Chart
APL_AXIS_THICKS	Axis Thickness in a Polar Chart
APL_LINE_FORMAT	Line Format to connect connecting lines
APL_LINES	Lines connecting polar lines show/hide

Series/Group Attributes

ATTRIBUTE	Description
ASG_LGND_ITEMS	Legend Items (#of items per row/column in legend area)
ASG_LGND_LAYOUT	Legend Layout (horizontal/vertial)
ASG_REVERSE_GROUPS	Reverse order of data for groups
ASG_REVERSE_SERIES	Reverse order of data for series
ASG_SERIES_COLOR	Series Color
ASG_SERIES_DELETE	Series Delete

ATTRIBUTE	Description
ASG_SWAP	Swap series and groups

Stock Chart Attributes

ATTRIBUTE	Description
ASM_HLWIDTH	High/Low bar Width of riser in a stock chart
ASM_ITEM_DTXT	Item Data Text for data text items in a stock market chart
ASM_METRIC1/2/3	Define metric lines in a stock chart
ASM_OCHEIGHT	Open/Close Height of ticks in a stock chart
ASM_OCWIDTH	Open/Close Width of ticks in a stock chart
ASM_SHOW_CLOSE	Show/Hide close ticks in a stock chart
ASM_SHOW_OPEN	Show/Hide Open ticks in a stock
ASM_SYMBOL_STYLE	Defines the style of Markers on Stock Chart

Table Chart Attributes

ATTRIBUTE	Description
ATC_ AUTOFIT_CELL	Autofit cell in a table chart
ATC_ AUTOFIT_COLHEAD	Autofit Column Header in a table chart
ATC_AUTOFIT_ALL	Autofit All (global autofitting) in a table chart
ATC_AUTOFIT_ROWHEAD	Autofit Row Header in a table chart
ATC_AUTOFIT_SUBJECT	Autofit Subject in a table chart
ATC_AUTOFIT_TEXT	Autofit Text in a table chart
ATC_DIVISIONS	Divisions in a table chart
ATC_LINECONTROL	Line Control in a table chart
ATC_UNIFORMCOLS	Uniform Columns in a table chart
ATC_UNIFORMROWS	Uniform Rows in a table chart

API Functions

- Initialization and Termination Functions
- Graph Structure Management Functions
- Graph Imaging Functions
- Draw Environment Functions
- Graph Modification Functions
- Selection List Management Functions
- Detection Node Management Functions
- Object Alignment Functions
- Memory Management Functions

- Annotation Functions
- Polygon Functions
- Platform-Independent File I/O
- RAM File Functions
- Getting/Setting Data
- Undo Functions
- Wireframe Functions
- Chart Layout Functions
- Translation Functions
- Export Functions
- Load/Save File Functions
- Advanced Drawing (OpenGL) Functions
- User Interface Dialog Functions
- Java Functions
- Headless-Server Functions for Unix Applications
- Miscellaneous Functions
- Data Callback Functions

Initialization and Termination Functions

The initialization and termination functions are only executed once in your application.

API Function Name	Description
InitTDGLIB()	Initialize the PGSDK library
FiniTDGLIB()	Finished Using PGSDK library

Graph Structure Management Functions

The graph structure management functions are executed once for each graph that your application draws. Memory must be allocated as part of the initialization process for each graph. The memory must be freed when your application is finished with the graph.

API Function Name	Description

AllocGraphPtr()	Allocate a Graph Pointer
FreeGraphPtr()	Free Graph Pointer

Graph Imaging Functions

The graph imaging functions are the ones that perform the actual drawing of the chart.

API Function Name	Description
DrawAnnotationLayer()	Draw annotation layer
DrawBackgroundLayer()	Draw background layer
DrawGraphLayer()	Draw Graph Layer
DrawTheGraph()	Draw the Graph
DryRunTheGraph()	Dry Run The Graph
IsGraphBackgroundVisible()	Is the graph background visible?

Draw Environment Functions

The draw environment functions are used to define the window, device context, destination rectangle, etc. where the graph(s) is(are) drawn.

API Function Name	Description
AllocDrawEnvPtr()	Allocate a Draw Environment Pointer
DEForceDetNodes()	Draw Environment Force Detection Nodes
DEGetDestRect ()	Draw Environment Get Destination Rectangle
DEGetDrawProcs ()	Draw Environment Get Drawing Procedures
DEGetGlobalFontInfo ()	Draw Environment Get Global Font Information
DEGetMeasurements()	Draw Environment Get Measurements
DEGetPenMode()	Draw Environment Get Pen Mode
DEGetPortInfo()	Draw Environment Get Port Information
DEGetVirtRect()	Draw Environment Get Virtual Rectangle
DERegisterCallbacks()	Draw Environment Register Callback functions
DESetActionProcs()	Draw Environment Set Action Procedures
DESetDestRect()	Draw Environment Set Destination Rectangle
DESetDrawProcs()	Draw Environment Set Drawing Procedures
DESetGlobalFontInfo()	Draw Environment Set Global Font Information
DESetMeasurements()	Draw Environment Set Measurements
DESetPenMode()	Draw Environment Set Pen Mode
DESetPortInfo()	Draw Environment Set Port Information
DESetVirtRect()	Draw Environment Set Virtual Rectangle
DrawBegin()	Draw Begin
DrawEnd()	Draw End
FreeDrawEnvPtr()	Free Draw Environment Pointer

Graph Modification Functions

The graph modification functions are primarily used to modify attributes of the chart. Your application can get information about the current attribute setting and also modify an attribute setting.

API Function Name	Description
GetGraphAttr()	Get Graph Attribute
GetGraphAttrSI()	Get Graph Attribute of an item in a selection list
GetGraphAttrSL()	Get Graph Attribute for all items in a selection list
GetObjectMobility()	Get Object Mobility
SetGraphAttr()	Set Graph Attribute
SetGraphAttrSI()	Set Graph Attribute for an item in a Selection List
SetGraphAttrSL()	Set Graph Attribute for all items in a Selection List

Selection List Management Functions

The selection list management functions provide the API calls necessary to create and edit selection lists. The selection lists contain information about each object the user selects for modification. These functions make it easy for your application to select objects and make modifications to the object attributes.

API Function Name	Description
CreateSelItemOutline()	Create Selection Item Outline
DestroySelItemOutline()	Destroy Selection Item Outline
GetFirstDetNodeRef()	Get First Detection Node Reference
IsMarkerSelected()	Is marker selected?
Item2DetRef()	Selection Item -to- Detection Reference conversion
MoveSelectedItem()	Move a selected item to a new virtual location
Remove()	Remove item(s) from a selection list
RemoveItem()	Remove an item from a selection list
Select_AddGroupsRelatedObjects()	Select Add Group-Related Objects to Selection List
Select_AddHeightRelatedObjects()	Select Add Height-Related Objects to Selection List
Select_AddItem()	Select Add Item to Selection List
Select_AddRelatedItems()	Select Add Related Items to Selection List
Select_AddSeriesRelatedObjects()	Select Add Series-Related Objects to Selection List
Select_AllocList()	Select Allocate Selection List
Select_CalcHandlesItem()	Select Calculate Handles Selection Item
Select_CalcHandlesList()	Select Calculate Handles Selection List
Select_ClearList()	Select Clear Selection List
Select_DrawXorItem()	Select Draw Xor Selection Item
Select_DrawXorList()	Select Draw Xor Selection List
Select_DuplicateList()	Select Duplicate Selection List

Select_FreeList()	Select Free Selection List
Select_GetFirstItem()	Select Get First Item in Selection List
Select_GetListBounds()	Select Get Selection List Bounds
Select_GetListInfo()	Select Get Selection List Information
Select_GetNextItem()	Select Get Next Item in Selection List
Select_GetSelectedSeriesID()	Select Get Selected Series ID
Select_Group()	Select Group in Selection List
Select_HandleHitTest()	Select Selection List Handle Hit Text
Select_IsEqual()	Select are two selection items equal?
Select_IsObjectSelected()	Select is object selected in selection list?
Select_IsSeriesRelatedObject()	Select is series-related object?
Select_RebuildItem()	Select rebuild item in Selection list
Select_RebuildList()	Select rebuild Selection list
Select_RemoveItem()	Select remove item from Selection list
Select_RemoveRelatedItems()	Select remove related items from Selection List
Select_SearchList()	Select Search Selection list
Select_SelectAllSeriesRelatedObjects()	Select all series-related objects in a chart
Select_UnGroup()	Select Ungroup Selection list
SelectDetNode()	Select Detection Node
SelectDetNodesInRect()	Select Detection Nodes in a Rectangle

Detection Node Management Functions

API Function Name	Description
FindDetNode()	Find Detection Node
FindNextDetNode()	Find Next Detection Node
FreeAllDetNodes()	Free All detection Nodes
FreeDetNode()	Free Detection Nodes
GetDetNode()	Get Detection Node
GetDetNodeObjectID()	Get Detection Node Object ID
GetNextDetNode()	Get Next Detection Node
GetRiserDetNodeLimit()	Get Riser Detection Node Limit
LockDetNodeRef()	Lock a Detection Node Reference
SetRiserDetNodeLimit()	Set Riser Detection Node Limit
UnlockDetNodeRef()	Unlock Detection Node Reference

Object Alignment Functions

The alignment functions make it easy to align objects on the screen.

API Function Name	Description
AlignInsideEdges()	Align Inside Edges of a selection list
AlignSelList_Bottom()	Align Selection List Bottom
AlignSelList_Center()	Align Selection List Center
AlignSelList_Left()	Align Selection List Left

The alignment functions make it easy to align objects on the screen.

API Function Name	Description
AlignSelList_Right()	Align Selection List Right
AlignSelList_Top()	Align Selection List Top
CenterOnPage()	Center objects in selection list
JoinHoriz()	Join movable objects in the selection list horizontally
JoinToFirst()	Join movable objects in the selection list to first
JoinVert()	Join movable objects in the selection list vertically
MakeSameSize()	Make all items in the selection list the same size as the first item
SizeToPicture()	Resize and relocate all items in the selection list

Memory Management Functions

API Function Name	Description
_SDK_AllocHandle()	Allocate relocatable memory in a platform-independent manner
_SDK_AllocHandleClear()	Allocate relocatable memory in a platform-independent manner and clear it
_SDK_AllocPtr()	Allocate non-relocatable memory
_SDK_AllocPtrClear()	Allocate non-relocatable memory and clear it
_xFreeHandle()	Free previously allocated relocatable memory
_xFreePtr()	Deallocate a block of non-relocatable memory
CloneHandle()	Clone a memory handle
ClonePtr()	Clone a Pointer
env_CompactMem()	Compact Memory
LockHandle()	Lock a memory handle
Ptr2Handle()	Pointer -to- Handle Conversion
Ptr2XHandle()	Pointer -to- XHandle Conversion
ReAllocHandle()	Reallocate Handle
ReAllocHandleClear()	Reallocate Handle Clear
SetvarHdlSize()	Set Variable Handle Size
SizeOfHandle()	Get Size of Handle
SizeOfPtr()	Get Size of Pointer
UnlockHandle()	Unlock memory handle

Annotation Functions

The annotation functions provide control for annotation objects:

API Function Name	Description
AnodeHdl2ID()	Convert Annotation Node Handle to Annotation ID
AnodeID2DetRef()	Convert Annotation Node ID to a Detection Reference
AnodeID2ZLayer()	Annotation Node ID -to- Z-Layer Conversion

The annotation functions provide control for annotation objects:

API Function Name	Description
Bring2Front()	Bring Selection List to Front
BringForward()	Bring Selection List Forward
CreateAnode()	Create an Annotation Node
CreatePictureAnode()	Create Picture Annotation Node
DetRef2AnodeID()	Detection Node Reference-to-Annotation Node ID conversion
DrawAllAnodes()	Draw all annotation nodes
DrawAnAnode()	Draw an annotation node
Duplicate()	Duplicate all annotation nodes
FreeAllAnodes()	Free All Annotation Nodes
GetAnnotationText()	Get Annotation Text
GetAnodeFromTextboxID()	Get Annotation Node from Text Box ID
GetRectAnodeInfo()	Get Rectangle Annotation Node Information
ID2AnodeHdl()	Annotation Node ID -to- Annotation Node Handle Conversion
IsAnode()	Is the object in at detection record an annotation node?
SelectAnode_AddItem()	Select Annotation Node Add Item to Selection List
SelectAnode_All()	Select All Annotation Nodes in Selection List
Send2Back()	Send Selection List to Back
SendBackward()	Send Selection List Backward
SetAnnotationText()	Set Annotation Text
SetRectAnodeInfo()	Set Rectangle Annotation Node Information
ZLayer2AnodeID()	Get annotation object ID of a given Z-layer

Polygon Functions

API Function Name	Description
AddPolyPoint()	Add Polygon Points
AllocPolyHandle()	Allocate a Polygon Handle
ClonePolyHandle()	Clone a Polygon Handle
ClosePolyHandle()	Close a Polygon Handle
dump_PolyHandle()	Output information about a polygon
FreePolyHandle()	Free Poly Handle
GetPolyBounds()	Get Polygon Bounds
GetPolyPoint()	Get Polygon Point
GetPolyPointCount()	Get Polygon Point Count
GetPolyPtrBounds()	Get Polygon Point Bounds
GetPolyPtrPointCount()	Get Polygon Point Count
LockPolyHandle()	Lock a polygon handle
Points2Poly()	Put points into a polygon
UnlockPolyHandle()	Unlock polygon handle

Platform-Independent File I/O

API Function Name	Description
IOclose()	Platform-independent file close
IOconnect()	Platform-independent file connect
IOgetc()	Platform-independent file get character
IOgets()	Platform-independent file string
IOopen()	Platform-independent file open
IOread()	Platform-independent file read
IOseek()	Platform-independent file seek
IOwrite()	Platform-independent file write

RAM File Functions

The RAM file functions can be used to save a chart to memory instead of a file.

API Function Name	Description
AllocRamFile()	Allocate a RAM File
FreeRamFile()	Free RAM File
GetRamFileBuffer()	Get RAM File Buffer
GetRamFileName()	Get RAM File Name
RFgetc()	Ram File Get Character
RFrewind()	Ram File Rewind
RFseek()	Ram File Seek
RFwrite()	Ram File Write
SetRamFileSize()	Set RAM File Size

Getting/Setting Data

API Function Name	Description
_SetGraphRowColData()	Set data in the spreadsheet cell for the graph
_SetGraphRowColPageData()	Set data in the spreadsheet cell for the graph
AccGraphDataStatus()	Verify a graph's data set
GetDataFormatID()	Get Data Format ID
GetDataRange()	Get Data Range
GetGraphDataInfo()	Get Graph Data Information
GetGraphFootNote()	Get Graph Footnote
GetGraphGroupsLabel()	Get Graph Groups Label
GetGraphGroupsTitle()	Get Graph Groups Title
GetGraphPage()	Get Graph Page
GetGraphRowColData()	Get Graph Row/Column Data
GetGraphRowColPageData()	Get Graph Row/Column Page Data
GetGraphRowColString()	Get Graph Row/Column String
GetGraphSeriesLabel()	Get Graph Series Label
GetGraphSeriesTitle()	Get Graph Series Title
GetGraphSubTitle()	Get Graph Subtitle

API Function Name	Description
GetGraphSubType()	Get Graph Sub-Type
GetGraphTitle()	Get Graph Title
GetGraphY1AxisTitle()	Get Graph Y1-Axis Title
GetGraphY2AxisTitle()	Get Graph Y2-Axis Title
GetGraphY3AxisTitle()	Get Graph Y3-Axis Title
GetGraphY4AxisTitle()	Get Graph Y4-Axis Title
GetGroupHeaderString()	Get Group Header String
GetRawLimits()	Get Raw Limits
GetSeriesHeaderString()	Get Series Header String
SetDataRange()	Set Data Range
SetGraphData()	Set Graph Data
SetGraphDataInfo()	Set Graph Data Information
SetGraphFootNote()	Set Graph Footnote
SetGraphGroupsLabel()	Set Graph Groups Label
SetGraphGroupsTitle()	Set Graph Groups Title
SetGraphLabels()	Set Graph Labels
SetGraphRowColData()	Set Graph Row/Column Data
SetGraphRowColString()	Set Graph Row/Column String
SetGraphSeriesLabel()	Set Graph Series Label
SetGraphSeriesTitle()	Set Graph Series Title
SetGraphSubTitle()	Set Graph Subtitle
SetGraphTitle()	Set Graph Title
SetGraphY1AxisTitle()	Set Graph Y1-Axis Title
SetGraphY2AxisTitle()	Set Graph Y2-Axis Title
SetGraphY3AxisTitle()	Set Graph Y3-Axis Title
SetGraphY4AxisTitle()	Set Graph Y4-Axis Title

UNDO Functions

The UNDO functions provide you with the ability to save user modifications and perform an UNDO on the modifications. Implemented in your application, these functions provide your user with the UNDO features expected in a Windows application.

API Function Name	Description
ClearUndo()	Clear the Undo List
GetUndoStatus()	Get Undo Status
IsUndoListEmpty()	Is the Undo List empty?
PerformUndo()	Perform Undo
SetUndoMode()	Set Undo Mode
SetUndoStatus()	Set Undo Status
UndoSelListMove()	Undo Selection List Move

3D Wireframe Functions

The 3D function draw, clear, and get information about the wire frame outline of the 3D chart to aid in rotating and scaling the chart:

API Function Name	Description
CheckWireframe()	Check the 3D graph wire frame
ClearWireframe()	Clear the 3D graph wire frame
DrawWireframe()	Draw 3D wire frame
GetWireFrameBounds()	Get Wire Frame Bounds

Chart Layout Functions

API Function Name	Description
AdjustToKeepChartOnScreen()	Adjust Frame to Keep Chart on Screen
AreVisualizeEffectUsed()	Are Visualize Effects Used?
CanUseEntireOrdAxis()	Can the chart use the entire ordinal axis to draw?
CreateRisers()	Create Risers
DetermineScale()	Determine Scale of the graph
EnablePlaceDefaultElements()	Routine to Disable/Enable PlaceDefaultElements Logic
GetAxisScaleMinMax()	Get the Minimum/Maximum Scaling for an Axis
GetGraphDocSize()	Get Graph Document Size
GetGraphOrientation()	Get Graph Orientation
GetGraphResolutionUnit()	Get Graph Resolution Units
GetGraphType()	Get Graph Type
GetGraphTypeIndex()	Get Graph Type Index
GetLegendLayoutInformation()	Get Legend Layout Information
GetSGCount()	Get Series/Group Count
GloSetMaxAutofitSize()	Global Set Maximum Autofit Size
GloSetMinAutofitSize()	Global Set Minimum Autofit Size
IsBipolar()	Is the graph a bi-polar chart?
IsDataText()	Is object data text?
IsDualYAxes()	Is the graph a Dual-Y axes chart?
IsO1AxisUsed()	Is the O1-Axis used in a chart?
IsXAxisUsed()	Is the X-Axis used in a chart?
IsY1AxisUsed()	Is the Y1-Axis used in a chart?
IsY2AxisUsed()	Is the Y2-Axis used in a chart?
IsY3AxisUsed()	Is the Y3-Axis used in a chart?
IsY4AxisUsed()	Is the Y4-Axis used in a chart?
PlaceDefaultElements	Place Default Elements
RecalcLegend()	Recalculate Legends
SetGraphDocSize()	Set Graph Document Side
SetGraphResolutionUnit()	Set Graph Resolution Units

Translation Functions

API Function Name	Description
AbsPoint()	Absolute Point
AddPoints()	Add Points
dCheckRect()	Verify the top, bottom, left and right parameters of a rectangle
dSetRect()	Define a Rectangle
dvCrust()	Device-to-Virtual Coordinates conversion of a CrustInfoRec
dvPoint()	Device-to-Virtual Coordinates conversion of a Point
dvPoint2D()	Device-to-Virtual Coordinates conversion of a Point2D
dvPoly()	Device-to-Virtual Coordinates conversion of a Poly
dvRect()	Device-to-Virtual Coordinates conversion of a Rect
dvScaleHeight()	Device-to-Virtual Coordinates conversion of Scale Height
dvScalePoint()	Device-to-Virtual Coordinates conversion of Scale Point
dvScalePoint2D()	Device-to-Virtual Coordinates conversion of Scale Point 2D
dvScaleWidth()	Device-to-Virtual Coordinates conversion of Scale Width
dvSlice()	Device-to-Virtual Coordinates conversion of a SliceInfoRec
dvWedge()	Device-to-Virtual Coordinates conversion of a WedgeInfoRec
pvScaleHeight()	Physical -to- Virtual Scale Height Conversion
pvScaleWidth()	Physical -to- Virtual Scale Width Conversion
ScalePoint()	Scale a Point
SubPoints()	Subtract Points
vCheckRect()	Verify rectangle definition is consistent with coordinate system
vdCrust	Virtual-to-Device Coordinates conversion of a CrustInfoRec
vdPoint()	Virtual-to-Device Coordinates conversion of a Point
vdPoint2D()	Virtual-to-Device Coordinates conversion of a Point2D
vdPoly()	Virtual-to-Device Coordinates conversion of a Poly
vdRect()	Virtual-to-Device Coordinates conversion of a Rect
vdScaleHeight()	Virtual-to-Device Coordinates conversion of Scale Height
vdScalePoint()	Virtual-to-Device Coordinates conversion of Scale Point

API Function Name	Description
vdScalePoint2D()	Virtual-to-Device Coordinates conversion of Scale Point2D
vdScaleWidth()	Virtual-to-Device Coordinates conversion of Scale Width
vdSlice()	Virtual-to-Device Coordinates conversion of a SliceInfoRec
vdWedge()	Virtual-to-Device Coordinates conversion of a WedgeInfoRec
vSetRect()	Set rectangle definition that is consistent with coordinate system
xBoxToRect()	Translate BoxInstRec to Rect
xCompareRects()	Translate compare rectangles
xDefineRect()	Translate define rectangle
xHeightOfRect()	Translate Height of Rectangle
xInsetRect()	Translate Inset Rectangle
xIntersectRect()	Translate Intersect Rectangle
xNormalizeDegrees()	Translate Normalize Degrees
xNormalizeRadians()	Translate Normalize Radians
xOffsetRect()	Translate Offset Rectangle
xPoint2DToPoint()	Translate Point2D -to- Point
xPointToPoint2D()	Translate Point -to- Point2D
xPt2Degrees()	Translate Point -to- Degrees
xPt2Radians()	Translate Point -to- Radians
xPtInEllipse()	Translate Point In Ellipse
xPtInPoly()	Translate Point In Polygon
xPtInRect()	Translate Point in Rectangle
xPtInSlice()	Determine if a point is in a slice
xPtInWedge()	Determine if a point is in a wedge
xRectToBox()	Translate Rectangle -to- BoxInstRec
xSumOfRects()	Translate Sum of Rectangles
xWidthOfRect()	Translate Width of Rectangle

Export Functions

API Function Name	Description
ExportBMP()	Export a chart image as a bit-map to a file
ExportBMPtoRamFile()	Export a chart image as a bit-map to a RAM file
ExportGIF()	Export a chart image as a .GIF file
ExportGIFtoMemory()	Export a chart image as a .GIF image to memory
ExportPNG()	Export a chart image as a PNG image to a file
ExportPNGToMemory()	Export a chart image as a PNG image to memory
SaveImageToFile()	Save the image of a graph to a file in any format
SaveImageToMemory()	Save the image of a graph to memory in any format

Load/Save File Functions

API Function Name	Description
Clone_TIFFGraph()	Clone a TIFF Graph
F3DF_ReadDescription()	Read description string from TIFF file
F3DF_ReadThumbnail()	Read thumbnail from TIFF file
Load_TIFFGraph()	Load a TIFF graph
Load_TIFFGraphPath()	Load a TIFF graph path
Save_TIFFGraph()	Save TIFF Graph
Save_TIFFGraphPath()	Save TIFF Graph Path
Save_TIFFGraphPathTrim()	Save TIFF Graph Path with trim option
ThumbWIN_GetFileDescriptionFromGraph()	Get File Description from Graph
ThumbWIN_GetThumbnailFromGraph()	Get Thumbnail from Graph
ThumbWIN_LoadDescription()	Load description into graph
ThumbWIN_LoadThumbnail()	Load thumbnail into graph
ThumbWIN_SetFileDescriptionIntoGraph()	Set File Description into Graph
ThumbWIN_UpdateInGraph()	Update Thumbnail in Graph

Advanced Drawing (OpenGL) Functions

API Function Name	Description
OGL_BeginFirstPass()	Open GL/Begin First Pass
OGL_BlitOffscreen()	Open GL/Blit Off-Screen
OGL_DrawBegin()	Open GL/Draw Begin
OGL_DrawEnd()	Open GL/Draw End
OGL_GetChartStatus()	Open GL/Get Chart Status
OGL_GetOffscreenDC()	Open GL/Get Off-Screen Device Context
OGL_InitOffscreen()	Open GL/Initialize Off-Screen
OGL_RegisterOffscreenPort()	Open GL/Register Off-Screen Port
OGL_RegisterPort()	Open GL/Register Port
OGL_SetTessellationDetail()	Open GL/Set Tessellation Detail

User Interface Dialog Functions

API Function Name	Description
InitAmigoDialog()	Initialize the Amigo user-interface dialogs
Prompt3DViewAnglePresetDialog()	Prompt 3D View Angle Preset Dialog
PromptChartOptionsDialog()	Prompt Chart Options Dialog
PromptChartTypesDialog()	Prompt Chart Types Dialog
PromptDataAnalyzerDialog()	Prompt Data Analyzer Dialog
PromptDropShadowDialog()	Prompt Drop Shadow Dialog
PromptFillGradientsDialog()	Prompt Fill Gradients Dialog
PromptFillPatternDialog()	Prompt Fill Pattern Dialog
PromptFillPicturesDialog()	Prompt Fill Pictures Dialog
PromptFillTexturesDialog()	Prompt Fill Textures Dialog
PromptFormattingDialog()	Prompt Formatting Dialog

API Function Name	Description
PromptGridScaleDialog()	Prompt Grids & Scales Dialog
PromptLightingPresetDialog()	Prompt Lighting Preset Dialog
PromptLineGradientsDialog()	Prompt Line Gradients Dialog
PromptLinePatternDialog()	Prompt Line Pattern Dialog
PromptLinePicturesDialog()	Prompt Line Pictures Dialog
PromptLineTexturesDialog()	Prompt Line Textures Dialog
PromptModelPresetDialog()	Prompt Line Model Preset Dialog
PromptSeriesOptionsDialog()	Prompt Line Series Options Dialog
PromptTextGradientsDialog()	Prompt Text Gradients Dialog
PromptTextPatternDialog()	Prompt Text Pattern Dialog
PromptTextPicturesDialog()	Prompt Text Pictures Dialog
PromptTextTexturesDialog()	Prompt Text Textures Dialog
PromptTitleDialog()	Prompt Titles Dialog
SetHelpFilePath()	Set Help File Path for User-Interface Dialogs

Java Functions

API Function Name	Description
ReadJAVAScriptChartFile	Read JAVA Script Chart File
WriteJAVAScriptChartFile	Write JAVA Script Chart File

Headless-Server Functions for Unix Applications

API Function Name	Description
GloGetDefaultFontName()	Get TrueType default font name
GloGetDefaultFontPath()	Get TrueType default font path
gloSetBackgroundGIF()	Set unpainted background pixels to be a passed-in GIF file.
GloSetDefaultFontName()	Set TrueType default font name
GloSetDefaultFontPath()	Set TrueType default font path
HS_DryRunTheGraph()	Headless-Server version of DryRunTheGraph()
HS_PlaceDefaultElements()	Headless-Server version of PlaceDefaultElements()

Miscellaneous Functions

API Function Name	Description
AllocDetListHandle()	Allocate a Detection List Handle
Categorize3DObject()	Is an object part of a graph cube, riser, or neither?
ChangeXanTextboxID()	Change Text Box ID
CheckDetList()	Check detection list in a draw environment
CreateMBCFontHandle()	Creates a font handle based on a existing logical font in the draw environment
EncompassLineThickness()	Expands the bounding rectangle of an item to include the thickness of its border or line
FindBoxForText()	Find a box for title, subtitle, or footnote text
FixupAxisTitleLocation()	Fix the axis title location

API Function Name	Description
FreeDetListHandle()	Free Detection Node List Handle
GetAdvFormat()	Get an Advanced Format structure
GetDLLVersion()	Get DLL Version String
GetGroupHeaderPos()	Get Group Header Position
GetRiserBounds()	Get Riser Bounds
GetRiserSurface()	Get Riser Surface
GetScaleBoundaries()	Get Scale Boundaries
GetSeriesHeaderPos()	Get Series Header Position
GetTableChartCellDimensions()	Get Table Chart Cell Dimensions
InitAdvFormat()	Initialize AdvFormat data structure
IsObjectATextBox()	Is the object a text box?
MBCExists()	Multi-Byte Characters Exist?
PackDate()	Pack date in the AdvFormat structure
RestoreVisualizeGraph()	Restore Visualize Graph
SetErrorCallBack()	Set Error Callback function
SetLineItemPoints()	Set Line Item Points
SetTableChartCellDimensions()	Set Table Chart Cell Dimensions
SetVisibilityMode()	Set Visibility Mode
StdFormat()	Format numerical values into a presentable string
SwapDetListHandle()	Swap Detection List Handle
Update_ClearList()	Update Clear List
Update_GetFirstItem()	Update Get First Item
Update_GetInvalRect()	Update Get Invalid Rectangle
Update_GetNextItem()	Update Get Next Item
UnPackDateTime()	Unpack Pack Date and Time in the AdvFormat structure

Data Callback Functions

The callback configuration of the DLL requires your application to provide a callback function. The function is called by the DLL with a row and column number to provide the data for charting. This configuration is best when your application is already buffering the data to be graphed. For example, your application collects the data from a database, a mainframe, a network, proprietary file format, etc., and buffers it for other purposes. Each time the graph is drawn the DLL calls your function to provide each data point and string for the chart. The following call-back functions are defined in the header file:

- `CALLBACK *LPFNGetDataCallBack`
- `CALLBACK *LPFNGetLabelCallBack`
- `CALLBACK *LPFNGetStringCallBack`

- `CALLBACK *LPFNSetPageCallBack`
- `CALLBACK *LPFNRealToStringCallBack`
- `CALLBACK *LPFNGetRawLimits`
- `CALLBACK *LPFNPictureCallBack`

This data interface calls a function in your application. A parameter points to a location for your function to load the requested data. The library does not buffer the data; once the data is charted, the data is overwritten with the next piece of data for charting. This data interface is best when your application already buffers the data. This prevents double buffering of the data by the library. Each time the chart is drawn by DrawTheGraph(), the data is requested from your application.

Compile and Link

The library includes a header file, PG32.H, contains all of the function prototypes, data structures, and constants. You must include the PG32.H file in all of your .C files that access a charting engine function:

```
/* Presentation Graphics SDK header file */
#include "pg32.h"
```

The compile command line must define WINDOWS and _WINDOWS. A sample compile command for Microsoft C 6 is shown below.

```
cl -AM -c -d -Gsw -Od -W2 -Zpi -DWINDOWS -D_WINDOWS
samp1.c
```

Setup for Compilers other than Visual C++ 6.0:

1) Add the \INCLUDE directory to your MSVC include path.

2) Add the \LIB directory to your MSVC lib path.

The Presentation Graphics SDK library must be linked into your application. A sample link command for Microsoft C 6 is shown below:

```
link samp1,/align:16,/map, pg mlibcew libw/NOD/NOE/CO,
samp1.def
```

What else do you need to know?

Three |D| Graphics has been licensing its technology to leading software companies such as Corel, Oracle, Cognos, and Symantec. We know that every developer has different needs and we aim to provide the highest quality technology and assistance possible. To learn more about how we can help you, please contact us at:

- Mailing Address: 11340 West Olympic Blvd., #300, Los Angeles, CA. 90064
- Telephone: 1.800.913.0008 or 1.310.231.3330
- FAX: 1.310.231.3303
- eMail: danw@threedgraphics.com
- WEB SITE: http://www.threedgraphics.com