

ProEssentials v1.5 Online Reference

[Function Groups](#)

[Alphabetical Function Listing](#)

[Property Groups](#)

[Alphabetic Property Listing](#)

[New Properties for Version 1.5](#)

[Constants and Enumerations](#)

[Messages](#)

[Structures](#)

[VBX Reference](#)

[Copyright](#)

Copyright Notice

Copyright 1994 Gigasoft, Inc. All Rights Reserved

No part of this manual may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine readable form without the prior written consent of Gigasoft, Inc.

Trademarks

ProEssentials is a trademark of Gigasoft, Inc. Gigasoft is pending federal registration as a trademark of Gigasoft, Inc.

Microsoft is a registered trademark of Microsoft Corporation. Windows and Visual Basic are trademarks of Microsoft Corporation.

Function Groups

Object Creation, Destruction. and Management

[PEcreate](#)
[PEdestroy](#)
[PEreinitialize](#)
[PEreinitializecustoms](#)
[PEresetimage](#)
[PEgetmeta](#)
[PEgethelpcontext](#)

Setting / Getting Properties

[PEvset](#)
[PEvget](#)
[PEnset](#)
[PEnget](#)
[PElset](#)
[PElget](#)
[PEszset](#)
[PEszget](#)
[PEvsetcell](#)
[PEvgetcell](#)
[PEsetglobal](#)
[PEgetglobal](#)
[PEsetgraph](#)
[PEgetgraph](#)
[PEsetsgraph](#)
[PEgetsgraph](#)
[PEsetpiechart](#)
[PEgetpiechart](#)

Built in Dialogs and Popup-Menu

[PElaunchcolordialog](#)
[PElaunchcustomize](#)
[PElaunchexport](#)
[PElaunchfontdialog](#)
[PElaunchmaximize](#)
[PElaunchpopupmenu](#)
[PElaunchprintdialog](#)
[PElaunchtextexport](#)

Exporting

[PEcopymetatoclipboard](#)
[PEcopymetatofile](#)
[PEcopybitmaptoclipboard](#)
[PEcopybitmaptofile](#)
[PEcopyoletoclipboard](#)

Serialization

[PEstore](#)

PEload

Alphabetic API Listing

[PEcopybitmaptoclipboard](#)

[PEcopybitmaptofile](#)

[PEcopymetatoclipboard](#)

[PEcopymetatofile](#)

[PEcopyoletoclipboard](#)

[PEcreate](#)

[PEdestroy](#)

[PEgetglobal](#)

[PEgetgraph](#)

[PEgethelpcontext](#)

[PEgetmeta](#)

[PEgetsgraph](#)

[PEgetpiechart](#)

[PElaunchcolordialog](#)

[PElaunchcustomize](#)

[PElaunchexport](#)

[PElaunchfontdialog](#)

[PElaunchmaximize](#)

[PElaunchpopupmenu](#)

[PElaunchprintdialog](#)

[PElaunchtextexport](#)

[PElget](#)

[PEload](#)

[PElset](#)

[PEnget](#)

[PEnset](#)

[PEreinitialize](#)

[PEreinitializecustoms](#)

[PEresetimage](#)

[PEsetglobal](#)

[PEsetgraph](#)

[PEsetpiechart](#)

[PEsetsgraph](#)

[PEstore](#)

[PEszget](#)

[PEszset](#)

[PEvget](#)

[PEvgetcell](#)

[PEvset](#)

[PEvsetcell](#)

PEcopybitmaptoclipboard

BOOL **PEcopybitmaptoclipboard** (*hObject*, *lpPoint*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
POINT FAR*	<i>lpPoint</i>	Pointer to POINT struct holding size.

This function places the current image, in the form of a bitmap, into the clipboard.

Parameter	Description
<i>lpPoint</i>	Pointer to a POINT struct where POINT.x stores the number of pixels wide and POINT.y stores the number of pixels high. The dimensions must be between 32 and 2000 pixels. To insure image quality, the aspect ratio (width/height) must be between 0.333 and 10.0.

Returns

NonZero if the function was successful, otherwise Zero.

PEcopybitmaptofile

BOOL **PEcopybitmaptofile** (*hObject*, *lpPoint*, *lpszFilename*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
POINT FAR*	<i>lpPoint</i>	Pointer to POINT struct holding size.
CHAR FAR*	<i>lpszFilename</i>	Pointer to null-terminated string.

This function places the current image, in the form of a bitmap, into a file named by *lpszFilename*.

Parameter	Description
<i>lpPoint</i>	Pointer to a POINT struct where POINT.x stores the number of pixels wide and POINT.y stores the number of pixels high. The dimensions must be between 32 and 2000 pixels. To insure image quality, the aspect ratio (width/height) must be between 0.333 and 10.0.
<i>lpszFilename</i>	This pointer must identify a valid filename.

Returns

NonZero if the function was successful, otherwise Zero.

PEcopymetatoclipboard

BOOL **PEcopymetatoclipboard** (*hObject*, *lpPoint*)

HWND *hObject* The handle returned from **PEcreate**.
POINT FAR* *lpPoint* Pointer to POINT struct holding size.

This function places the current image, in the form of a metafile, into the clipboard.

Parameter	Description
<i>lpPoint</i>	Pointer to a POINT struct where POINT.x stores the width and POINT.y stores the height. The dimensions are in 1/100th of a millimeter. The dimensions must not equal zero. To insure image quality, the aspect ratio (width/height) must be between 0.333 and 10.0.

Returns

NonZero if the function was successful, otherwise Zero.

PEcopymetatofile

BOOL **PEcopymetatofile** (*hObject, lpPoint, lpszFilename*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
POINT FAR*	<i>lpPoint</i>	Pointer to POINT struct holding size.
CHAR FAR*	<i>lpszFilename</i>	Pointer to a null-terminated string naming file.

This function places the current image, in the form of a metafile, into a file named by *lpszFilename*.

Parameter	Description
<i>lpPoint</i>	Pointer to a POINT struct where POINT.x stores the width and POINT.y stores the height. The dimensions are in 1/100th of a millimeter. The dimensions must not equal zero. To insure image quality, the aspect ratio (width/height) must be between 0.333 and 10.0.

<i>lpszFilename</i>	This pointer must identify a valid filename.
---------------------	--

Returns

NonZero if the function was successful, otherwise Zero.

PEcopyoletoclipboard

BOOL **PEcopyoletoclipboard** (*hObject*, *lpPoint*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
POINT FAR*	<i>lpPoint</i>	Pointer to POINT struct holding size.

This function places the current image, in the form of an OLE object, into the clipboard.

Parameter	Description
<i>lpPoint</i>	Pointer to a POINT struct where POINT.x stores the width and POINT.y stores the height. The dimensions are in 1/100th of a millimeter. The dimensions must not equal zero. To insure image quality, the aspect ratio (width/height) must be between 0.333 and 10.0.

Returns

NonZero if the function was successful, otherwise Zero.

PEcreate

HWND **PEcreate** (*nObjectType*, *dwStyle*, *lpRect*, *hParent*, *nID*)

UINT	<i>nObjectType</i>	Specifies the type of object to create
DWORD	<i>dwStyle</i>	Specifies window styles
RECT FAR*	<i>lpRect</i>	Location and size of object
HWND	<i>hParent</i>	Parent window of object
UINT	<i>nID</i>	Child window identifier

This function allocates memory for the object, creates the window, and registers the windows handle with ProEssentials DLL.

Parameter	Description
<i>nObjectType</i>	PECONTROL_GRAPH Graph Object PECONTROL_SGRAPH Scientific Graph Object PECONTROL_PIE Pie Chart Object

<i>dwStyle</i>	WS_TABSTOP WS_VISIBLE WS_DISABLED WS_GROUP
----------------	--

Combine the above Windows defined styles with the bitwise-or operator. The function automatically adds the WS_CHILD, WS_VSCROLL, and WS_HSCROLL styles, then passes this value to the Windows **CreateWindow** function.

lpRect Passed to the Windows **CreateWindow** function.

hParent Passed to the Windows **CreateWindow** function.

nID Passed to the Windows **CreateWindow** function.

Returns

This function returns the handle which was assigned by the Windows operating system. If the function could not allocate memory, or **CreateWindow** fails, the function returns zero.

Comments

The return value from this function is used in both Windows API and ProEssentials API calls. This return value must not be lost as long as the object remains in memory. Storing the return value in a global variable is required.

PEdestroy must be used to destroy the window and free its memory and resources.

PEdestroy

BOOL **PEdestroy** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function destroys the window associated with this object, frees memory and resources, and unregisters the handle with the ProEssentials DLL.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

If this function is not called once for each time a **PEcreate** function is called, there will be objects stored in memory until the DLL is unloaded. When the DLL is unloaded, it will destroy any abandoned objects remaining in memory.

PEgetglobal

BOOL **PEgetglobal** (*hObject*, *lpGlbProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
GLOBALPROPERTIES FAR*	<i>lpGlbProps</i>	Points to a structure.

This function fills all members of the GLOBALPROPERTIES structure by copying properties from the object identified by *hObject*. This structure contains all the common non-array properties that belong to all ProEssentials controls.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use **PEgetgraph**, **PEgetsgraph**, or **PEgetpiechart** to fill the specific structures for the Graph, Scientific Graph, and PieChart controls. Use the **PEvget** function to get property arrays such as PEP_faYDATA.

PEgetgraph

BOOL **PEgetgraph** (*hObject*, *lpGrphProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
GRAPHPROPERTIES FAR*	<i>lpGrphProps</i>	Points to a structure.

This function fills all members of the GRAPHPROPERTIES structure by copying properties from the object identified by *hObject*. This structure contains the non-array properties that belong to the Graph control.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEgetglobal** function to get global properties. Use the **PEvget** function to get property arrays such as PEP_faYDATA.

PEgethelpcontext

DWORD **PEgethelpcontext** (*hWnd*)

HWND *hWnd* Handle of window to get context for.

This function returns the help-context assigned to the window identified by *hWnd*.

Returns

Zero if the window identified by *hWnd* is not a ProEssentials based window, otherwise the help-context to be used with "PEGRAPHS.HLP".

Example

In response to the F1 key being pressed.

```
DWORD     wHelpID;
HWND     hWnd;

/* get window with current focus */
hWnd = GetFocus();

/* check to see if it is a ProEssentials Object */
wHelpID = PEgethelpcontext (hWnd);

if (wHelpID)
{
    WinHelp(hMainFrame, "pegraphs.hlp", HELP_CONTEXT, wHelpID);
}
else
{
    /* a non-ProEssentials controls has the focus */
}
```

PEgetmeta

HMETAFILE **PEgetmeta** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function returns the handle of a metafile that is managed by the ProEssentials object identified by *hObject*. Do not use the Windows **DeleteMetafile** function on this handle.

Returns

Retrieves the object's main metafile handle (the one it paints the object's image with.)

PEgetsgraph

BOOL **PEgetsgraph** (*hObject*, *lpSGrphProps*)

HWND	<i>hObject</i>	Handle from PEcreate .
SGRAPHPROPERTIES FAR*	<i>lpSGrphProps</i>	Pointer to a structure.

This function fills all members of the SGRAPHPROPERTIES structure by copying properties from the object identified by *hObject*. This structure holds the non-array properties that belong to the Scientific Graph Object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEgetglobal** function to get global properties. Use the **PEvget** function to get property arrays such as PEP_faYDATA.

PEgetpiechart

BOOL **PEgetpiechart** (*hObject*, *lpPieProps*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
PIEPROPERTIES FAR*	<i>lpPieProps</i>	Pointer to a structure.

This function fills all members of the PIEPROPERTIES structure by copying properties from the object identified by *hObject*. This structure holds the non-array properties that belong to the PieChart Object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEgetglobal** function to get global properties. Use the **PEvget** function to get property arrays such as PEP_faYDATA.

PElaunchcolordialog

BOOL **PElaunchcolordialog** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function invokes a modal color customization dialog. Depending on the current viewing style the dialog will either adjust the color-version or monochrome-version color set.

Returns

If successful, the function returns the result of the closing dialog, either IDOK, or IDCANCEL.

If unsuccessful, the function returns -1.

PElaunchcustomize

INT **PElaunchcustomize** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function invokes a modal customization dialog which manages the object's customizable properties.

Returns

If successful, the function returns the result code of the closing dialog. Possible values are:

IDCANCEL
IDOK
IDEXPORTBUTTON
IDMAXIMIZEBUTTON
IDORIGINALBUTTON

If unsuccessful, the function returns -1.

PElaunchexport

BOOL **PElaunchexport** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function invokes a modal export dialog.

Returns

NonZero if the function was successful, otherwise Zero.

PElaunchfontdialog

INT **PElaunchfontdialog** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function invokes a modal font customization dialog.

Returns

If successful, the function returns the result of the closing dialog, either IDOK, or IDCANCEL.

If unsuccessful, the function returns -1.

PElaunchmaximize

BOOL **PElaunchmaximize** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function invokes a maximized modal dialog with a copy of the current object maximized to fit inside it.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

If **PEP_bNOSCROLLINGSUBSETCONTROL** equals TRUE, the maximized version of the object will reset this property, allowing access to subset parameter customizations.

PElaunchpopupmenu

BOOL **PElaunchpopupmenu** (*hObject*, *pLocation*)

HWND	<i>hObject</i>	The handle returned from <u>PEcreate</u> .
POINT FAR*	<i>pLocation</i>	Specifies coordinates where popup is placed.

This function invokes a popup-menu at the location specified. The coordinates defined in *pLocation* are with respect to the objects client area.

Returns

NonZero if the function was successful, otherwise Zero.

PElaunchprintdialog

BOLD **PElaunchprintdialog**(*hObject*, *bFullPage*, *lpPoint*)

HWND	<i>hObject</i>	The handle returned from <u>PEcreate</u> .
BOOL	<i>bFullPage</i>	Controls page size.
POINT FAR*	<i>lpPoint</i>	Pointer to POINT struct holding size.

This function invokes a modal print dialog. If *bFullPage* equals TRUE then *lpPoint* should be set to NULL. If *bFullPage* equals FALSE then *lpPoint* must point to a POINT struct that holds the size (in 1/100th millimeter units) of the image to export.

Returns

NonZero if the function was successful, otherwise Zero.

PElaunchtextexport

BOOL **PElaunchtextexport** (*hObject*, *bToFile*, *lpszFilename*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
BOOL	<i>bToFile</i>	File or Clipboard destination.
CHAR FAR*	<i>lpszFilename</i>	Pointer to a Null terminated string.

This function invokes a modal text-export dialog. The dialog allows the user to export the objects data to either the clipboard or a file.

Parameter	Description
<i>bToFile</i>	TRUE causes the data to be sent to the clipboard. FALSE causes the data to be sent a file whose name is specified by <i>lpszFilename</i> .
<i>lpszFilename</i>	This pointer must identify a valid filename.

Returns

NonZero if the function was successful, otherwise Zero.

PEIget

LONG **PEIget** (*hObject*, *nProperty*)

** 16 bit specific

HWND

hObect

The handle returned from **PEcreate**.

UINT

nProperty

Identifies the property.

This function gets object properties by returning the data selected by *hObject*, and *nProperty*. This function works with color properties.

Returns

If successful the function returns the property requested. If an error occurs the function returns -1.

Comments

Version 1.0 used this function to access float properties. Version 1.5 now uses [Pevget](#) to get float properties.

Refer to [Appendix B Property Reference](#) to view possible property codes.

PEload

BOOL **PEload** (*hObject*, *lphGlobal*)

HWND	<i>hObject</i>	The handle returned from <u>PEcreate</u> .
HGLOBAL FAR*	<i>lphGlobal</i>	Handle to global memory containing data.

This function reloads an objects data from binary form located in a global memory object.

Parameter	Description
<i>lphGlobal</i>	Pointer to a global memory handle. This memory object holds binary information which was stored with the <u>PEstore</u> function.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Call the **PEreinitialize** and **PEresetimage** functions to complete the objects transformation.

Use **PEstore** to initially store the object's information into the binary format.

PEIset

BOOL **PEIset** (*hObject, nProperty, IData*) ** 16 bit specific

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies which property is set.
LONG	<i>IData</i>	Data which will be set.

This function sets object properties located by *hObject* and *nProperty*. This function works with color properties.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Version 1.0 used this function to access float properties. Version 1.5 now uses [PEvset](#) to get float properties.

Refer to [Appendix B Property Reference](#) to view possible property codes.

PEnget

INT **PEnget** (*hObject*, *nProperty*)

HWND	<i>hObect</i>	The handle returned from <u>PEcreate</u> .
UINT	<i>nProperty</i>	Identifies the property.

This function gets object properties by returning the data selected by *hObject*, and *nProperty*. This function works with all properties which are two bytes in length (integer, boolean, and handle.)

Returns

If successful the function returns the property requested. If an error occurs the function returns -1.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

PEset

BOOL **PEset** (*hObject, nProperty, nData*)

HWND	<i>hObect</i>	The handle returned from <u>PEcreate</u> .
UINT	<i>nProperty</i>	Identifies which property is set.
INT	<i>nData</i>	Data which will be set.

This function sets object properties located by *hObject* and *nProperty*. This function works with all properties which are two bytes in length (integer, boolean, and handle.)

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

PEreinitialize

BOOL **PEreinitialize** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function resets the object to new properties and data.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

After setting any pre-initialization properties or calling **PEload**, this function resets those object variables which are pre-calculated to optimize object performance.

You must also call **PEresetimage** to complete the objects transformation.

This function calls **PEreinitializecustoms** so you dont have to call both **PEreinitialize** and **PEreinitializecustoms**.

PEreinitializecustoms

BOOL **PEreinitializecustoms** (*hObject*)

HWND *hObject* The handle returned from **PEcreate**.

This function resets the object after customizable properties have been adjusted. The object manages an Original parameter set, and a Custom parameter set. Depending on the property **PEP_bCUSTOM**, this function establishes the working parameters which are directly involved in producing the object's image. This function must be called after altering any Original or Custom parameter.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

You must also call **PEresetimage** to complete the objects transformation.

PEresetimage

BOOL **PEresetimage** (*hObject*, *nXAspect*, *nYAspect*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
INT	<i>nXAspect</i>	The width aspect element.
INT	<i>nYAspect</i>	The height aspect element.

This function is called to reset the image when: 1) the images aspect ratio changes, and 2) the objects properties are adjusted.

Parameter	Description
<i>nXAspect</i>	Relative width. If zero then this function uses the object's current aspect ratio.
<i>nYAspect</i>	Relative height. If zero then this function uses the object's current aspect ratio.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

The values *nXAspect* and *nYAspect* are developer defined. The ratio of *nXAspect* divided by *nYAspect* is what controls the image creation process. *nXAspect* = 1, *nYAspect* = 2 will produce the same image as *nXAspect* = 100, *nYAspect* = 200.

PEsetglobal

BOOL **PEsetglobal** (*hObject*, *lpGlbProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
GLOBALPROPERTIES FAR*	<i>lpGlbProps</i>	Points to a structure.

This function copies the properties contained in the GLOBALPROPERTIES structure to the object identified by *hObject*. This structure holds all the common non-array properties that belong to ProEssentials objects.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use **PEsetgraph**, **PEsetsgraph**, or **PEsetpiechart** to set the specific property structures for the Graph, Scientific Graph, and PieChart controls. Use the **PEvset** function to set property arrays such as PEP_faYDATA.

PEsetgraph

BOOL **PEsetgraph** (*hObject*, *lpGrphProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
GRAPHPROPERTIES FAR*	<i>lpGrphProps</i>	Points to a structure.

This function copies the properties contained in the GRAPHPROPERTIES structure to the Graph Object identified by *hObject*. This structure holds the non-array properties that belong to the Graph Object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEsetglobal** function to set global properties. Use the **PEvset** function to set property arrays such as PEP_faYDATA.

PEsetsgraph

BOOL **PEsetsgraph** (*hObject*, *lpSGrphProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
SGRAPHPROPERTIES FAR*	<i>lpSGrphProps</i>	Points to a structure.

This function copies the properties contained in the SGRAPHPROPERTIES structure to the Scientific Graph Object identified by *hObject*. This structure holds the non-array properties that belong to the Scientific Graph Object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEsetglobal** function to set global properties. Use the **PEvset** function to set property arrays such as PEP_faYDATA.

PEsetpiechart

BOOL **PEsetpiechart** (*hObject*, *lpPieProps*)

HWND	<i>hObject</i>	The handle from PEcreate .
PIEPROPERTIES FAR*	<i>lpPieProps</i>	Points to a structure.

This function copies the properties contained in the PIEPROPERTIES structure to the Pie Chart Object identified by *hObject*. This structure holds the non-array properties that belong to the PieChart Object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Use the **PEsetglobal** function to set global properties. Use the **PEvset** function to set property arrays such as PEP_faYDATA.

PEstore

BOOL **PEstore** (*hObject*, *lphGlobal*, *lpdwSize*)

HWND	<i>hObject</i>	The handle returned from PEcreate .
HGLOBAL FAR*	<i>lphGlobal</i>	Handle to destination global object.
DWORD FAR*	<i>lpdwSize</i>	Size of information transferred.

This function stores the objects data in a binary form for later retrieval.

Parameter	Description
<i>lphGlobal</i>	Pointer to a global memory handle. If the function is successful, <i>lphGlobal</i> will point to a handle of a global memory object which was created by the ProEssentials DLL. The calling program is responsible for destroying this global memory object when finished with the data.
<i>lpdwSize</i>	Pointer to a DWORD. If the function is successful, <i>lpdwSize</i> will point to a DWORD which holds the number of bytes transferred into the global memory object.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

To reload the information use **PEload** to transfer the information into an object of the same type.

PEszget

BOOL **PEszget** (*hObject*, *nProperty*, *lpszDestination*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies the property.
CHAR FAR*	<i>lpszDestination</i>	Pointer to buffer where string will be stored.

This function gets object string properties by copying the string selected by *hObject* and *nProperty* into the memory location pointed to by *lpszDestination*.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

This function can be used to get tab delimited string property arrays, but make sure that the buffer pointed to by *lpszDestination* is a large enough to receive data.

PEszset

BOOL **PEszset** (*hObject*, *nProperty*, *lpszData*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies the property.
CHAR FAR*	<i>lpszData</i>	Pointer to string data.

This function sets object string properties by copying the string pointed to by *lpszData* into the object property located by *hObject* and *nProperty*.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

This function can be used to set tab delimited string property arrays. It will attempt to set PEP_nSUBSET subsetlabels, and PEP_nPOINTS pointlabels.

PEvget

INT **PEvget** (*hObject*, *nProperty*, *lpvDestination*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies the property.
VOID FAR*	<i>lpvDestination</i>	Pointer to buffer where property will be stored.

This function gets object property arrays and properties of non-simple data type (including float properties) by copying the data selected by *hObject* and *nProperty* into the memory location pointed to by *lpvDestination*.

Returns

If *lpvDestination* equals NULL and *nProperty* represents a property array, the function returns the number of bytes needed to store the data transferred if *lpvDestination* is not equal to NULL.

If *lpvDestination* is not equal to NULL, the function returns Zero if successful, otherwise, the function returns -1.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

PEvgetcell

BOOL **PEvgetcell** (*hObject*, *nProperty*, *nCell*, *lpvDestination*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies the property.
UINT	<i>nCell</i>	Identifies the cell (element) of array.
VOID FAR*	<i>lpvDestination</i>	Pointer to buffer where property will be stored.

This function gets individual elements of object property arrays by copying the array element selected by *hObject*, *nProperty*, and *nCell* into the memory location pointed to by *lpvDestination*.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to [Appendix B Property Reference](#) to view possible property codes.

Make sure that the buffer pointer to by *lpvDestination* is large enough to hold the property received. Individual string elements will never exceed 48 characters in length.

PEvset

BOOL **PEvset** (*hObject*, *nProperty*, *lpvData*, *nItems*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies which property is set.
VOID FAR*	<i>lpvData</i>	Pointer to data which will be copied.
UINT	<i>nItems</i>	Number of elements (for property arrays.)

This function sets object property arrays and properties of non-simple data type (including float properties) by copying the data pointed to by *lpvData* into the object property located by *hObject* and *nProperty*.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to the [Appendix B Property Reference](#) to view possible property codes.

When used with PEP_szaSUBSETLABELS and PEP_szaPOINTLABELS, the tab delimited string information pointed to by *lpvData* will be corrupted during the de-tokenization process. **Do not** pass tab delimited string array information which has static scope.

PEvsetcell

BOOL **PEvsetcell** (*hObject*, *nProperty*, *nCell*, *lpvData*)

HWND	<i>hObect</i>	The handle returned from PEcreate .
UINT	<i>nProperty</i>	Identifies which property is set.
UINT	<i>nCell</i>	Identifies the cell (element) of array.
VOID FAR*	<i>lpvData</i>	Pointer to data which will be copied.

This function sets individual elements of object property arrays by copying the data pointed to by *lpvData* into the object property array element located by *hObject*, *nProperty*, and *nCell*.

Returns

NonZero if the function was successful, otherwise Zero.

Comments

Refer to the [Appendix B Property Reference](#) to view possible property codes.

Property Groups

Global Properties

PEP_b3DDIALOGS

PEP_bALLOWCUSTOMIZATION

PEP_bALLOWEXPORTING

PEP_bALLOWMAXIMIZATION

PEP_bALLOWPOPOP

PEP_bALLOWSUBSETHOTSPOTS

PEP_bALLOWUSERINTERFACE

PEP_bAUTOIMAGERESET

PEP_bCUSTOM

PEP_nDATAPRECISION

PEP_bDATASHADOWS

PEP_nDEFORIENTATION

PEP_dwDESKCOLOR

PEP_bDIALOGSHOWN

PEP_bDIRTY

PEP_nFONTSIZE

PEP_dwGRAPHBACKCOLOR

PEP_dwGRAPHFORECOLOR

PEP_structHOTSPOTDATA

PEP_bLABELBOLD

PEP_szLABELFONT

PEP_bLABELITALIC

PEP_bLABELUNDERLINE

PEP_rectLOGICALLOC

PEP_szMAINTITLE

PEP_bMAINTITLEBOLD

PEP_szMAINTITLEFONT

PEP_bMAINTITLEITALIC

PEP_bMAINTITLEUNDERLINE

PEP_hMEMBITMAP

PEP_hMEMDC

PEP_dwMONODESKCOLOR

PEP_dwMONOGRAPHBACKCOLOR

PEP_dwMONOGRAPHFORECOLOR

PEP_dwMONOSHADOWCOLOR

PEP_dwMONOTABLEBACKCOLOR

PEP_dwMONOTABLEFORECOLOR

PEP_dwMONOTEXTCOLOR

PEP_bMONOWITHSYMBOLS

PEP_nOBJECTTYPE

PEP_nPAGEHEIGHT

PEP_nPAGEWIDTH

PEP_nPOINTS

PEP_bPREPAREIMAGES

PEP_dwSHADOWCOLOR

PEP_dwaSUBSETCOLORS

PEP_szaSUBSETLABELS

PEP_nSUBSETS

PEP_dwaSUBSETSHADES

PEP_szSUBTITLE

PEP_bSUBTITLEBOLD

PEP_szSUBTITLEFONT
PEP_bSUBTITLEITALIC
PEP_bSUBTITLEUNDERLINE
PEP_dwTEXTCOLOR
PEP_nVIEWINGSTYLE
PEP_dwWDESKCOLOR
PEP_dwWGRAPHBACKCOLOR
PEP_dwWGRAPHFORECOLOR
PEP_dwWSHADOWCOLOR
PEP_dwWTEXTCOLOR
PEP_faYDATA

Graph and Scientific Graph Properties

PEP_bALLOWBESTFITCURVE
PEP_bALLOWBESTFITLINE
PEP_bALLOWCOORDPROMPTING
PEP_bALLOWDATAHOTSPOTS
PEP_bALLOWGRAPHHOTSPOTS
PEP_bALLOWLINE
PEP_bALLOWPLOTCUSTOMIZATION
PEP_bALLOWPOINT
PEP_bALLOWPOINTSPLUSLINE
PEP_bALLOWPOINTSPLUSSPLINE
PEP_bALLWSPLINE
PEP_nALLOWZOOMING
PEP_nBESTFITDEGREE
PEP_nCOMPARISONSUBSETS
PEP_nCURVEGRANULARITY
PEP_bFORCERIGHTYAXIS
PEP_bGRIDINFRONT
PEP_nGRIDLINECONTROL
PEP_nINITIALSCALEFORRYDATA
PEP_nINITIALSCALEFORRYDATA
PEP_szLOWERBOUNDTEXT
PEP_fLOWERBOUNDVALUE
PEP_fMANUALMAXRY
PEP_fMANUALMAXY
PEP_fMANUALMINRY
PEP_fMANUALMINY
PEP_fMANUALRYAXISLINE
PEP_fMANUALRYAXISTICK
PEP_bMANUALRYAXISTICKNLINE
PEP_nMANUALSCALECONTROLRY
PEP_bMANUALSCALECONTROLRY
PEP_fMANUALYAXISLINE
PEP_fMANUALYAXISTICK
PEP_bMANUALYAXISTICKNLINE
PEP_bMARKDATAPOINTS
PEP_bNEGATIVEFROMXAXIS
PEP_bNOSCROLLINGSUBSETCONTROL
PEP_fNULLDATAVALUE
PEP_nPLOTINGMETHOD
PEP_nPOINTSIZE
PEP_naRANDOMSUBSETSTOGRAPH
PEP_nRYAXISCOMPARISONSUBSETS
PEP_szRYAXISLABEL

PEP_nRYAXISSCALECONTROL
PEP_nSCALEFORRYDATA
PEP_nSCALEFORRYDATA
PEP_nSCROLLINGSUBSETS
PEP_naSUBSETLINETYPES
PEP_naSUBSETPOINTTYPES
PEP_nSYMBOLFREQUENCY
PEP_btREATCOMPSASNORMAL
PEP_szUPPERBOUNDTEXT
PEP_fUPPERBOUNDVALUE
PEP_nVBOUNDARYTYPES
PEP_szXAXISLABEL
PEP_szYAXISLABEL
PEP_nYAXISSCALECONTROL

Graph Specific Properties

PEP_bALLOWAREA
PEP_bALLOWBAR
PEP_bALLOWBESTFITCURVEII
PEP_bALLOWBESTFITLINEII
PEP_bALLOWHISTOGRAM
PEP_bALLOWPOINTHOTSPOTS
PEP_bALLOWTABLEHOTSPOTS
PEP_nALTFREQTHRESHOLD
PEP_naALTFREQUENCIES
PEP_szaAPPENDPOINTLABELDATA
PEP_bAPPENDTOEND
PEP_faAPPENDYDATA
PEP_naAUTOSTATSUBSETS
PEP_bFORCEVERTICALPOINTS
PEP_nGRAPHPLUSTABLE
PEP_szMANUALMAXDATASTRING
PEP_szMANUALMAXPOINTLABEL
PEP_fMANUALSTACKEDMAXY
PEP_nMAXPOINTSTOGRAPH
PEP_bNORANDOMPOINTSTOGRAPH
PEP_bNOSTACKEDDATA
PEP_szaPOINTLABELS
PEP_nPOINTSTOGRAPH
PEP_nPOINTSTOGRAPHINIT
PEP_nPOINTSTOGRAPHVERSION
PEP_naRANDOMPOINTSTOGRAPH
PEP_nSPECIFICPLOTMODE
PEP_dwTABLEBACKCOLOR
PEP_szTABLEFONT
PEP_dwTABLEFORECOLOR
PEP_nTABLEWHAT
PEP_nTARGETPOINTSTOTABLE
PEP_dwWTABLEBACKCOLOR
PEP_dwWTABLEFORECOLOR

Scientific Graph Specific Properties

PEP_bALLOWBUBBLE
PEP_nALLOWDATALABELS
PEP_bALLOWSTICK
PEP_nBUBBLESIZE

PEP_szaDATAPOINTLABELS
PEP_bGRAPHDATA LABELS
PEP_nINITIALSCALEFORXDATA
PEP_fMANUALMAXX
PEP_fMANUALMINX
PEP_bMANUALSCALECONTROLX
PEP_fMANUALXAXISLINE
PEP_fMANUALXAXISTICK
PEP_bMANUALXAXISTICKNLINE
PEP_nSCALEFORXDATA
PEP_nXAXISSCALECONTROL
PEP_faXDATA
PEP_faZDATA
PEP_fZOOMMAXX
PEP_fZOOMMAXY
PEP_fZOOMMINX
PEP_fZOOMMINY
PEP_bZOOMMODE

PieChart Specific Properties

PEP_bALLOWPOINTHOTSPOTS
PEP_nAUTOEXPLODE
PEP_nDATA LABELTYPE
PEP_nGROUPINGPERCENT
PEP_szaPOINT LABELS
PEP_faXDATA

Alphabetic Property Listing

PEP_b3DDIALOGS
PEP_bALLOWAREA
PEP_bALLOWBAR
PEP_bALLOWBESTFITCURVE
PEP_bALLOWBESTFITCURVEII
PEP_bALLOWBESTFITLINE
PEP_bALLOWBESTFITLINEII
PEP_bALLOWBUBBLE
PEP_bALLOWCOORDPROMPTING
PEP_bALLOWCUSTOMIZATION
PEP_bALLOWDATAHOTSPOTS
PEP_nALLOWDATALABELS
PEP_bALLOWEXPORTING
PEP_bALLOWGRAPHHOTSPOTS
PEP_bALLOWHISTOGRAM
PEP_bALLOWLINE
PEP_bALLOWMAXIMIZATION
PEP_bALLOWPLOTCUSTOMIZATION
PEP_bALLOWPOINT
PEP_bALLOWPOINTHOTSPOTS
PEP_bALLOWPOINTSPLUSLINE
PEP_bALLOWPOINTSPLUSSPLINE
PEP_bALLOWPOPOP
PEP_bALLOWSPLINE
PEP_bALLOWTABLEHOTSPOTS
PEP_bALLOWSTICK
PEP_bALLOWSUBSETHOTSPOTS
PEP_bALLOWUSERINTERFACE
PEP_nALLOWZOOMING
PEP_nALTFREQTHRESHOLD
PEP_naALTFREQUENCIES
PEP_szaAPPENDPOINTLABELDATA
PEP_bAPPENDTOEND
PEP_faAPPENDYDATA
PEP_nAUTOEXPLODE
PEP_bAUTOIMAGERESET
PEP_naAUTOSTATSUBSETS
PEP_nBESTFITDEGREE
PEP_nBUBBLESIZE
PEP_nCOMPARISONSUBSETS
PEP_nCURVEGRANULARITY
PEP_bCUSTOM
PEP_nDATALABELTYPE
PEP_szaDATAPOINTLABELS
PEP_nDATAPRECISION
PEP_bDATASHADOWS
PEP_nDEFORIENTATION
PEP_dwDESKCOLOR
PEP_bDIALOGSHOWN
PEP_bDIRTY
PEP_nFONTSIZE
PEP_bFORCERIGHTYAXIS

PEP_bFORCEVERTICALPOINTS
PEP_dwGRAPHBACKCOLOR
PEP_bGRAPHDATALABELS
PEP_dwGRAPHFORECOLOR
PEP_nGRAPHPLUSTABLE
PEP_bGRIDINFRONT
PEP_nGRIDLINECONTROL
PEP_nGROUPINGPERCENT
PEP_structHOTSPOTDATA
PEP_nINITIALSCALEFORRYDATA
PEP_nINITIALSCALEFORXDATA
PEP_nINITIALSCALEFORYDATA
PEP_bLABELBOLD
PEP_szLABELFONT
PEP_bLABELITALIC
PEP_bLABELUNDERLINE
PEP_rectLOGICALLOC
PEP_szLOWERBOUNDTEXT
PEP_fLOWERBOUNDVALUE
PEP_szMAINTITLE
PEP_bMAINTITLEBOLD
PEP_szMAINTITLEFONT
PEP_bMAINTITLEITALIC
PEP_bMAINTITLEUNDERLINE
PEP_szMANUALMAXDATASTRING
PEP_fMANUALMAXRY
PEP_fMANUALMAXX
PEP_fMANUALMAXY
PEP_szMANUALMAXPOINTLABEL
PEP_fMANUALMINRY
PEP_fMANUALMINX
PEP_fMANUALMINY
PEP_fMANUALRYAXISLINE
PEP_fMANUALRYAXISTICK
PEP_bMANUALRYAXISTICKNLINE
PEP_bMANUALSCALECONTROLRY
PEP_bMANUALSCALECONTROLX
PEP_bMANUALSCALECONTROLRY
PEP_fMANUALSTACKEDMAXY
PEP_fMANUALXAXISLINE
PEP_fMANUALXAXISTICK
PEP_bMANUALXAXISTICKNLINE
PEP_fMANUALYAXISLINE
PEP_fMANUALYAXISTICK
PEP_bMANUALYAXISTICKNLINE
PEP_bMARKDATAPOINTS
PEP_nMAXPOINTSTOGRAPH
PEP_hMEMBITMAP
PEP_hMEMDC
PEP_dwMONODESKCOLOR
PEP_dwMONOGRAPHBACKCOLOR
PEP_dwMONOGRAPHFORECOLOR
PEP_dwMONOSHADOWCOLOR
PEP_dwMONOTABLEBACKCOLOR
PEP_dwMONOTABLEFORECOLOR
PEP_dwMONOTEXTCOLOR

PEP_bMONOWITHSYMBOLS
PEP_bNEGATIVEFROMXAXIS
PEP_bNORANDOMPOINTSTOGRAPH
PEP_bNOSCROLLINGSUBSETCONTROL
PEP_bNOSTACKEDDATA
PEP_fNULLDATAVALUE
PEP_nOBJECTTYPE
PEP_nPAGEHEIGHT
PEP_nPAGEWIDTH
PEP_nPLOTTINGMETHOD
PEP_szaPOINTLABELS
PEP_nPOINTS
PEP_nPOINTSIZ
PEP_nPOINTSTOGRAPH
PEP_nPOINTSTOGRAPHINIT
PEP_nPOINTSTOGRAPHVERSION
PEP_bPREPAREIMAGES
PEP_naRANDOMPOINTSTOGRAPH
PEP_naRANDOMSUBSETSTOGRAPH
PEP_nRYAXISCOMPARISONSUBSETS
PEP_szRYAXISLABEL
PEP_nRYAXISSCALECONTROL
PEP_nSCALEFORXDATA
PEP_nSCALEFORRYDATA
PEP_nSCALEFORRYDATA
PEP_nSCROLLINGSUBSETS
PEP_dwSHADOWCOLOR
PEP_nSPECIFICPLOTMODE
PEP_dwaSUBSETCOLORS
PEP_szaSUBSETLABELS
PEP_nSUBSETS
PEP_naSUBSETLINETYPES
PEP_naSUBSETPOINTTYPES
PEP_dwaSUBSETSHADES
PEP_szSUBTITLE
PEP_bSUBTITLEBOLD
PEP_szSUBTITLEFONT
PEP_bSUBTITLEITALIC
PEP_bSUBTITLEUNDERLINE
PEP_nSYMBOLFREQUENCY
PEP_dwTABLEBACKCOLOR
PEP_szTABLEFONT
PEP_dwTABLEFORECOLOR
PEP_nTABLEWHAT
PEP_nTARGETPOINTSTOTABLE
PEP_dwTEXTCOLOR
PEP_bTREATCOMPSASNORMAL
PEP_szUPPERBOUNDTEXT
PEP_fUPPERBOUNDVALUE
PEP_nVBOUNDARYTYPES
PEP_nVIEWINGSTYLE
PEP_dwWDESKCOLOR
PEP_dwWGRAPHBACKCOLOR
PEP_dwWGRAPHFORECOLOR
PEP_dwWSHADOWCOLOR
PEP_dwWTABLEBACKCOLOR

PEP_dwWTABLEFORECOLOR
PEP_dwWTEXTCOLOR
PEP_szXAXISLABEL
PEP_nXAXISSCALECONTROL
PEP_faXDATA
PEP_szYAXISLABEL
PEP_nYAXISSCALECONTROL
PEP_faYDATA
PEP_faZDATA
PEP_fZOOMMAXX
PEP_fZOOMMAXY
PEP_fZOOMMINX
PEP_fZOOMMINY
PEP_bZOOMMODE

New Properties for V1.5

Global Properties

PEP_bALLOWSUBSETHOTSPOTS
PEP_bAUTOIMAGERESET
PEP_structHOTSPOTDATA

Graph and Scientific Graph Properties

PEP_bALLOWBESTFITCURVE
PEP_bALLOWBESTFITCURVEII
PEP_bALLOWBESTFITLINE
PEP_bALLOWBESTFITLINEII
PEP_bALLOWCOORDPROMPTING
PEP_bALLOWDATAHOTSPOTS
PEP_bALLOWGRAPHHOTSPOTS
PEP_bALLOWLINE
PEP_bALLOWPLOTCUSTOMIZATION
PEP_bALLOWPOINT
PEP_bALLOWPOINTSPLUSLINE
PEP_bALLOWPOINTSPLUSSPLINE
PEP_bALLOWSPLINE
PEP_nALLOWZOOMING
PEP_nBESTFITDEGREE
PEP_nCURVEGRANULARITY
PEP_bFORCERIGHTYAXIS
PEP_nINITIALSCALEFORRYDATA
PEP_fMANUALMAXRY
PEP_fMANUALMINRY
PEP_fMANUALRYAXISLINE
PEP_fMANUALRYAXISTICK
PEP_bMANUALRYAXISTICKNLINE
PEP_nMANUALSCALECONTROLRY
PEP_fMANUALYAXISLINE
PEP_fMANUALYAXISTICK
PEP_bMANUALYAXISTICKNLINE
PEP_bMARKDATAPOINTS
PEP_bNEGATIVEFROMXAXIS
PEP_fNULLDATAVALUE
PEP_nPOINTSIZE
PEP_nRYAXISCOMPARISONSUBSETS
PEP_szRYAXISLABEL
PEP_nRYAXISSCALECONTROL
PEP_nSCALEFORRYDATA
PEP_naSUBSETLINETYPES
PEP_naSUBSETPOINTTYPES
PEP_nSYMBOLFREQUENCY

Graph Specific Properties

PEP_bALLOWAREA
PEP_bALLOWBAR
PEP_bALLOWHISTOGRAM
PEP_bALLOWPOINTHOTSPOTS
PEP_bALLOWTABLEHOTSPOTS
PEP_nALTFREQTHRESHOLD

PEP_naALTFREQUENCIES
PEP_szaAPPENDPOINTLABELDATA
PEP_bAPPENDTOEND
PEP_faAPPENDYDATA
PEP_szMANUALMAXDATASTRING
PEP_szMANUALMAXPOINTLABEL
PEP_fMANUALSTACKEDMAXY
PEP_nMAXPOINTSTOGRAPH
PEP_bNORANDOMPOINTSTOGRAPH
PEP_nSPECIFICPLOTMODE
PEP_nTARGETPOINTSTOTABLE

Scientific Graph Specific Properties

PEP_bALLOWBUBBLE
PEP_nALLOWDATALABELS
PEP_bALLOWSTICK
PEP_nBUBBLESIZE
PEP_szaDATAPOINTLABELS
PEP_fMANUALXAXISLINE
PEP_fMANUALXAXISTICK
PEP_bMANUALXAXISTICKNLINE
PEP_faZDATA
PEP_fZOOMMAXX
PEP_fZOOMMAXY
PEP_fZOOMMINX
PEP_fZOOMMINY
PEP_bZOOMMODE

PieChart Specific Properties

PEP_bALLOWPOINTHOTSPOTS
PEP_nAUTOEXPLODE

PEP_b3DDIALOGS

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	3dDialogs

Purpose

This property controls whether dialogs invoked from the object utilize 3D or 2D controls.

Setting	Description
TRUE	Dialogs use 3D controls.
FALSE	Dialogs use 2D controls.

PEP_bALLOWAREA

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowArea

Purpose

This property controls whether the Area plotting method is accessible.

Setting	Description
TRUE	Area Graph is available.
FALSE	Area Graph is not available.

PEP_bALLOWBAR

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowBar

Purpose

This property controls whether the Bar plotting method is accessible.

Setting	Description
TRUE	Bar Graph is available.
FALSE	Bar Graph is not available.

PEP_bALLOWBESTFITCURVE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowBestFitCurve

Purpose

This property controls whether the Points plus Best Fit Curve plotting method is accessible.

Setting

TRUE
FALSE

Description

Best Fit Curve Graph is available.
Best Fit Curve Graph is not available.

Comments

The PEP_nBESTFITDEGREE property is used to control the degree of polynomial to be generated.

PEP_bALLOWBESTFITCURVEII

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowBestFitCurveII

Purpose

This property controls whether the Points plus Best Fit Curve II plotting method is accessible. This property produces a best fit curve for only those points currently graphed.

Setting

Description

TRUE	Best Fit Curve Graph II is available.
FALSE	Best Fit Curve Graph II is not available.

Comments

The PEP_nBESTFITDEGREE property is used to control the degree of polynomial to be generated.

PEP_bALLOWBESTFITLINE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowBestFitLine

Purpose

This property controls whether the Points plus Best Fit Line plotting method is accessible.

Setting

TRUE
FALSE

Description

Best Fit Line Graph is available.
Best Fit Line Graph is not available.

PEP_bALLOWBESTFITLINEII

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowBestFitLineII

Purpose

This property controls whether the Points plus Best Fit Line II plotting method is accessible. This property produces a best fit line for only those points currently graphed.

Setting

TRUE
FALSE

Description

Best Fit Line Graph II is available.
Best Fit Line Graph II is not available.

PEP_bALLOWBUBBLE

Scope	Scientific Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowBubble

Purpose

This property controls whether the Bubble plotting method is accessible.

Setting

TRUE
FALSE

Description

Bubble Graph is available.
Bubble Graph is not available.

Comments

The PEP_nBUBBLESIZE property is used to control the size of the maximum bubble drawn.

The PEP_faZDATA property is used to control the size of individual bubbles.

PEP_bALLOWCOORDPROMPTING

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowCoordPrompting

Purpose

This property controls whether the user will receive coordinate feedback when they double-click the graphs grid. When the grid area is double-clicked, a small popup window captures the mouse and displays the clicked coordinates with precision defined by PEP_nDATAPRECISION.

Setting

TRUE
FALSE

Description

Enable auto-coordinate prompting.
Disable auto-coordinate prompting.

PEP_bALLOWCUSTOMIZATION

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowCustomization

Purpose

This property controls whether the user can access the Customization Dialog.

Setting

Description

TRUE	User can access Customization Dialog.
FALSE	User can not access Customization Dialog.

Comments

The object can still be customized via keyboard and popup-menu interfaces. Use PEP_bALLOWUSERINTERFACE and PEP_bALLOWPOPUP to further control object customization access.

PEP_bALLOWDATAHOTSPOTS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowDataHotSpots

Purpose

This property controls whether data-points will be Hot-Spots. The developer can then respond when the user clicks or double-clicks a data-point.

Setting	Description
TRUE	Enable data hot-spots.
FALSE	Disable data hot-spots.

Comments

SDK developers, see [HOTSPOTDATA](#) for more information.
VBX developers, see [DataHotSpot](#) for more information

The Graph Object has a maximum of 250 data point Hot-Spots. If the amount of points being currently graphed is larger than 250, then there will be no data Hot-Spots. The user can reduce the number of points being graphed in order to enable data point Hot-Spots.

The Scientific Graph Object has no limit on the number of data point Hot-Spots.

PEP_nALLOWDATALABELS

Scope	Scientific Graph Object.
Type	UINT
Default	PEADL_DATAVALUES
Custom	NA
VBX	AllowDataLabels

Purpose

This property controls what type of data-point labels can be placed next to data-points.

Setting

PEADL_NONE

PEADL_DATAVALUES

PEADL_POINTLABELS

PEADL_DATAPOINTLABELS

Description

Disable data-point label capabilities.

Numerical value of data-point.

Uses PEP_szaPOINTLABELS.

Uses PEP_szaDATAPOINTLABELS.

Comments

PEP_bGRAPHDATALABELS is used control whether data-point labels are currently graphed.

PEP_bALLOWEXPORTING

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowExporting

Purpose

This property controls whether the user will have access to export capabilities.

Setting	Description
TRUE	User can access export capabilities.
FALSE	User can not access export capabilities.

Comments

Export capabilities consist of the Export Dialog, Print Dialog, and Text/Data-Export Dialog.

PEP_bALLOWGRAPHHOTSPOTS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowGraphHotSpots

Purpose

This property controls whether the graphs grid will be Hot-Spots. The developer can then respond when the user clicks or double-clicks the grid portion of the object.

Setting	Description
TRUE	Enable graph hot-spots.
FALSE	Disable graph hot-spots.

Comments

SDK developers, see [HOTSPOTDATA](#) for more information.
VBX developers, see [GraphHotSpot](#) for more information

PEP_bALLOWHISTOGRAM

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowHistogram

Purpose

This property controls whether the Histogram plotting method is accessible.

Setting	Description
TRUE	Histogram is available.
FALSE	Histogram is not available.

PEP_bALLOWLINE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowLine

Purpose

This property controls whether the Line plotting method is accessible.

Setting	Description
TRUE	Line Graph is available.
FALSE	Line Graph is not available.

PEP_bALLOWMAXIMIZATION

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowMaximization

Purpose

This property controls whether the user will have access to maximizing the object.

Setting	Description
TRUE	User can maximize object.
FALSE	User can not maximize object.

PEP_bALLOWPLOTCUSTOMIZATION

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowPlotCustomization

Purpose

This property controls whether the user will have access to changing the objects plotting method.

Setting

TRUE
FALSE

Description

User can change plotting method.
User can not change plotting method.

PEP_bALLOWPOINT

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowPoint

Purpose

This property controls whether the Point plotting method is accessible.

Setting	Description
TRUE	Point Graph is available.
FALSE	Point Graph is not available.

PEP_bALLOWPOINTHOTSPOTS

Scope	Graph and Pie Chart Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowPointHotSpots

Purpose

This property controls whether point labels will be Hot-Spots. The developer can then respond when the user clicks or double-clicks a point label.

Setting	Description
TRUE	Enable point label hot-spots.
FALSE	Disable point label hot-spots.

Comments

SDK developers, see [HOTSPOTDATA](#) for more information.
VBX developers, see [PointHotSpot](#) for more information

PEP_bALLOWPOINTSPLUSLINE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowPointsPlusLine

Purpose

This property controls whether the Points Plus Line plotting method is accessible.

Setting

TRUE
FALSE

Description

Points Plus Line Graph is available.
Point Plus Line Graph is not available.

PEP_bALLOWPOINTSPLUSPLINE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowPointsPlusSpline

Purpose

This property controls whether the Points Plus Spline plotting method is accessible.

Setting

TRUE
FALSE

Description

Points Plus Spline Graph is available.
Point Plus Spline Graph is not available.

PEP_bALLOWPOPUP

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowPopup

Purpose

This property controls whether the user will have access to the objects popup-menu.

Setting

TRUE
FALSE

Description

User can access popup menu.
User can not access popup menu.

PEP_bALLOWSPLINE

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowSpline

Purpose

This property controls whether the Points plus Spline and Spline plotting methods are accessible.

Setting	Description
TRUE	Spline Graph is available.
FALSE	Spline Graph is not available.

PEP_bALLOWSTICK

Scope	Scientific Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowStick

Purpose

This property controls whether the Stick plotting method is accessible.

Setting	Description
TRUE	Stick Graph is available.
FALSE	Stick Graph is not available.

PEP_bALLOWSUBSETHOTSPOTS

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowSubsetHotSpots

Purpose

This property controls whether subset labels will be Hot-Spots. The developer can then respond when the user clicks or double-clicks a subset label.

Setting	Description
TRUE	Enable subset label hot-spots.
FALSE	Disable subset label hot-spots.

Comments

SDK developers, see [HOTSPOTDATA](#) for more information.
VBX developers, see [SubsetHotSpot](#) for more information

PEP_bALLOWTABLEHOTSPOTS

Scope	Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	AllowTableHotSpots

Purpose

This property controls whether the Graphs table portion will be a Hot-Spot. The developer can then respond when the user clicks or double-clicks items in the table.

Setting	Description
TRUE	Enable table hot-spots.
FALSE	Disable table hot-spots.

Comments

SDK developers, see [HOTSPOTDATA](#) for more information.

VBX developers, see [TableHotSpot](#) for more information

PEP_bALLOWUSERINTERFACE

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AllowUserInterface

Purpose

This property controls whether the object translates keyboard and mouse commands.

Setting

TRUE
FALSE

Description

User can access user interface.
User can not access user interface.

Comments

This property also controls how the object handles WM_GETDLGCODE. If this property is TRUE, then the control will allow only ProEssentials interface keystrokes. If this property is FALSE, then the control will accept all keystrokes. This allows the developer to implement their own interface.

PEP_nALLOWZOOMING

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEAZ_NONE
Custom	NA
VBX	AllowZooming

Purpose

This property controls the zooming capabilities of the object.

The following constants apply to this property:

Constant	Description
PEAZ_NONE	Disable zooming capabilities.
PEAZ_HORIZONTAL	Enable horizontal zooming.
PEAZ_VERTICAL	Enable vertical zooming.
PEAZ_HORZANDVERT	Enable horizontal and vertical zooming.

Comments

Other related properties include PEP_bZOOMMODE, PEP_fZOOMMINX, PEP_fZOOMMAXX, PEP_fZOOMMINY, PEP_fZOOMMAXY

PEP_nALTFREQTHRESHOLD

Scope	Graph Object.
Type	UINT
Default	25
Custom	NA
VBX	AltFreqThreshold

Purpose

This property represents the threshold number of PEP_nPOINTSTOGRAPH which triggers PEP_naALTFREQUENCIES to be utilized in determined an alternate frequency for placing point labels and tabled data.

Comments

For more information see PEP_naALTFREQUENCIES.

PEP_naALTFREQUENCIES

Scope	Graph Object.
Type	Property Array of UINTs
Default	NA
Custom	NA
VBX	AltFrequencies

Purpose

This property array is used to define meaningful frequencies to place point labels and tabled data.

Comments

The Graph Object has a mechanism to produce legible point labels and tabled data when many points are being graphed. This mechanism is implemented with the following properties:

PEP_naALTFREQUENCIES, PEP_nALTFREQTHRESHOLD, and PEP_nTARGETPOINTSTOTABLE.

PEP_nALTFREQTHRESHOLD is used to trigger this mechanism.

PEP_nTARGETPOINTSTOTABLE represents the number of point labels which will be attempted to be displayed.

PEP_naALTFREQUENCIES holds an array of meaningful frequencies.

*Note: You are not required to define alternate frequencies for this mechanism to function. Only define alternate frequencies if you want the frequency selection process to pick a frequency which seems logical with respect to the type of data being displayed. If no alt frequencies are defined, the Graph Object can pick any frequency which produces a number of point labels closest to PEP_nTARGETPOINTSTOTABLE.

Example:

You are displaying point information as seconds and have 2400 seconds worth of data. You could define alternate frequencies of 5, 15, 30, 60, and 120, or 5 seconds, 15 seconds, 1/2 minute, 1 minute, and 2 minutes. As the user utilizes the Customization Dialog and changes the PEP_nPOINTSTOGRAPH property, the Graph Object would select a frequency by dividing PEP_nPOINTSTOGRAPH by PEP_nTARGETPOINTSTOTABLE.

C / C++ Example

```
int nArray[5];
nArray[0] = 5;
nArray[1] = 15;
nArray[2] = 30;
nArray[3] = 60;
nArray[4] = 120;
PEvset (hWndPE, PEP_naALTFREQUENCIES, nArray, 5);
```

VBX Example

```
PEGraph1.AltFrequencies(0) = 5
PEGraph1.AltFrequencies(1) = 15
PEGraph1.AltFrequencies(2) = 30
PEGraph1.AltFrequencies(3) = 60
PEGraph1.AltFrequencies(4) = 120
```

PEP_szaAPPENDPOINTLABELDATA

Scope	Graph Object.
Type	Property Array of Tab Delimited Strings
Default	empty
Custom	NA
VBX	NA

Purpose

This property is for transferring new point labels into a Graph Object which is being used in a real-time implementation. Existing point labels will first be shifted to make room for the new labels. Each string in the tab delimited array should be less than or equal to 48 characters in length.

Comments

This property is write-only and can only be used with [PEvset](#). This property must be set before PEP_faAPPENDYDATA. For more information on the use of this property see [PEP_faAPPENDYDATA](#).

C / C++ Example

```
/* 4 new tab delimited point labels */  
char szPointLabels[] = 56:04\t56:05\t56:06\t56:07\t;  
PEvset (hWndPE, PEP_szaAPPENDPOINTLABELDATA, szPointlabels, 4);
```

VBX Example

You are required to call PEvset for real-time implementations.

```
t$ = Chr$(9) ** tab character **  
L$ = 56:04+t$+56:05+t$+56:06+t$+56:07+t$  
test% = PEvset (PEGraph1, PEP_szaAPPENDPOINTLABELDATA, ByVal L$, 4)
```

PEP_bAPPENDTOEND

Scope	Graph Object.
Type	BOOL
Default	TRUE
Custom	NA
VBX	AppendToEnd

Purpose

This property is only needed in a real-time implementation. It controls whether the graph automatically scrolls left or right.

Setting

Description

TRUE	Data is appended at end. Graph scrolls right to left.
FALSE	Data is appended at beginning. Graph scrolls left to right.

Comments

For more information on real-time implementations, see [PEP_faAPPENDYDATA](#).

PEP_faAPPENDYDATA

Scope	Graph Object.
Type	Property Array of type FLOAT
Default	empty
Custom	NA
VBX	NA

Purpose

This property is for transferring new YData into a Graph Object which is being used in a real-time implementation. Existing YData will first be shifted to make room for the new data.

Comments

This property is write-only and can only be used with PEvset.

After this property is set, the Graph Object will automatically regenerate and display the new image based upon the new data transferred into the object. This is why PEP_szaAPPENDPOINTLABELDATA should be first set before using this property. **Do not** call **PEreinitialize** or **PEresetimage**.

Important, when using PEvset, the argument *nItems* should represent the number of elements **per subset**. If the object has 4 subsets and you are adding 1 element of YData per subset, *nItems* should equal 1, and the array located by *lpvData* should contain 4 elements. The Graph Object knows how many subsets its managing, so it only needs to know how many new items of YData per subset is being added. If more than one piece of YData is being appended per subset, the data should be prepared in the following form:

Data Format

s1p1	s1p2	s1p3	s1p4	...	s1pj
s2p1	s2p2	s2p3	s2p4	...	s2pj
s3p1	s3p2	s3p3	s3p4	...	s3pj
...
sip1	sip2	sip3	sip4	...	sipj

C / C++ Example

```
float fData[4]; /* first place new data into fData */
/* 4 subsets * 1 point = 4 elements, yet nItems = 1 */
PEvset (hWndPE, PEP_faAPPENDYDATA, fData, 1);
```

VBX Example

You are required to call PEvset for real-time implementations.

```
ReDim fData(4) ** first place new data into fData **
** 4 subsets * 1 point = 4 elements, yet nItems = 1 **
test% = PEvset (PEGraph1, PEP_faAPPENDYDATA, fData(0), 1)
```

PEP_nAUTOEXPLODE

Scope	Pie Chart Object.
Type	UINT
Default	FALSE
Custom	NA
VBX	AutoExplode

Purpose

This property controls whether the Pie Chart will automatically explode pie slices when a point-label is double clicked.

Constant

PEAE_NONE

PEAE_ALLSUBSETS

PEAE_INDSUBSETS

Description

Disable automatic slice exploding.

Enable automatic slice exploding. Slices are exploded for all subsets.

Enable automatic slice exploding. Slices are exploded for individual subsets only.

Comments

Enabling this property will force PEP_bALLOWPOINTHOTSPOTS to TRUE.

PEP_bAUTOIMAGERESET

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	NA

Purpose

This property controls whether the ProEssentials will automatically call PEreinitialize if properties were adjusted and then an image is requested.

Setting	Description
TRUE	Enable automatic image resetting.
FALSE	Disable automatic image resetting.

Comments

This property should only be disabled in certain real-time implementation. This allows new data to be set without having to reinitialize the object.

If this property is enabled, the ProEssentials will monitor when properties are set. If properties are adjusted and then an image is requested prior to a call to PEreinitialize or PEreinitialzecustoms, the ProEssentials will automatically force a call to PEreinitialize. Note, this is not desirable. Normally, the developer should always call PEreinitialize or PEreinitialzecusoms after setting properties.

PEP_naAUTOSTATSUBSETS

Scope	Graph Object.
Type	Property Array of UINTs
Default	No automatic subsets
Custom	NA
VBX	AutoStatSubsets

Purpose

This property array is used to tell the object to automatically generate comparison subsets and append the newly generated subsets to the objects data.

The type of automatic data to generate is defined by the following constants:

Constant	Description
PEAS_SUMPP	Sum all subsets per point
PEAS_MINAP	Minimum all subsets all points
PEAS_MINPP	Minimum all subsets per point
PEAS_MAXAP	Maximum all subsets all points
PEAS_MAXPP	Maximum all subsets per point
PEAS_AVGAP	Average all subsets all points
PEAS_AVGPP	Average all subsets per point
PEAS_P1SDAP	Average plus 1 Std Dev all points
PEAS_P1SDPP	Average plus 1 Std Dev per point
PEAS_P2SDAP	Average plus 2 Std Devs all points
PEAS_P2SDPP	Average plus 2 Std Devs per point
PEAS_P3SDAP	Average plus 3 Std Devs all points
PEAS_P3SDPP	Average plus 3 Std Devs per point
PEAS_M1SDAP	Average minus 1 Std Dev all points
PEAS_M1SDPP	Average minus 1 Std Dev per point
PEAS_M2SDAP	Average minus 2 Std Devs all points
PEAS_M2SDPP	Average minus 2 Std Devs per point
PEAS_M3SDAP	Average minus 3 Std Devs all points
PEAS_M3SDPP	Average minus 3 Std Devs per point
PEAS_PARETO_ASC	Pareto ascending accumulated percent line
PEAS_PARETO_DEC	Pareto descending accumulated percent line

Comments

This is a pre-initialization property. During the **PEreinitialize** function, the object will use the information in this property array to generate the requested comparison data and then destroy the property array information.

If invalid constants are supplied to the PEP_naAUTOSTATSUBSETS property, those subsets will be ignored.

Standard Deviation calculations are only practical when the data set is sufficiently large. For all point calculations, subsets * points ≥ 15 , and for per point calculations, subsets ≥ 15 .

Also note that the above constants that end with AP (for all points) will not be included into table sections. They will only be included into the graph section since all point values are identical.

The following examples cause the Graph Object to generate the all-point average and the per-point average comparison subsets.

C / C++ Example

```
int nArray[2];  
nArray[0] = PEAS_AVGAP;  
nArray[1] = PEAS_AVGPP;  
PEvset (hWndPE, PEP_naAUTOSTATSUBSETS, nArray, 2);
```

VBX Example

```
PEGraph1.AutoStatSubsets(0) = PEAS_AVGAP  
PEGraph1.AutoStatSubsets(1) = PEAS_AVGPP
```

PEP_nBESTFITDEGREE

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEBFD_2ND
Custom	NA
VBX	BestFitDegree

Purpose

This property controls the degree of polynomial which is produced when the plotting method is Points plus Best Fit Curve.

The following constants apply to this property:

<u>Constant</u>	Description
PEBFD_2ND	Second degree polynomial.
PEBFD_3RD	Third degree polynomial.
PEBFD_4TH	Fourth degree polynomial.

PEP_nBUBBLESIZE

Scope	Scientific Graph Object.
Type	UINT
Default	PEBS_MEDIUM
Custom	NA
VBX	BubbleSize

Purpose

This property controls the overall size of bubbles produced when the plotting method is Bubble.

The following constants apply to this property:

<u>Constant</u>	Description
PEBS_SMALL	Small bubbles.
PEBS_MEDIUM	Medium bubbles.
PEBS_LARGE	Large bubbles.

PEP_nCOMPARISONSUBSETS

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	empty
Custom	NA
VBX	ComparisonSubsets

Purpose

This property is used to establish custom (developer defined) comparison subsets.

Comments

This is a pre-initialization property. During the **PEreinitialize** function, the object will use this property to designate the last PEP_nCOMPARISONSUBSETS subset indexes as comparison data and then sets this property to zero.

Custom comparison subsets are always located after non-comparison subsets and before auto-stat subsets.

It is the developers responsibility to transfer the comparison data (along with non-comparison data) into the object via the PEP_faXDATA, PEP_faYDATA, and PEP_szaSUBSETLABELS properties, and precede the respective subset labels with an asterisks (which identifies the subset as a comparison subset.)

PEP_nCURVEGRANULARITY

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PECG_MEDIUM
Custom	NA
VBX	CurveGranularity

Purpose

This property controls the quantity of line-segments used to produce the curved lines for the Spline and Best Fit Curve plotting methods.

The following constants apply to this property:

<u>Constant</u>	Description
PECG_COARSE	Fewest line-segments.
PECG_MEDIUM	Medium number of line-segments.
PECG_FINE	Most line-segments.

Comments

If you expect sharp curves in your data, then choosing PECG_FINE will produce better printed curves. If your data is not expected to have sharp curves, then choosing PECG_COARSE will draw the fastest curves.

PEP_bCUSTOM

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NA

Purpose

This property defines whether the object displays the Original or Custom parameter set.

Setting	Description
TRUE	Create image using Custom parameter set.
FALSE	Create image using Original parameter set.

Comments

If this property is adjusted, then the **PEreinitializecustoms** and **PEresetimage** functions need to be called in order to complete the transformation.

PEP_nDATALABELTYPE

Scope	Pie Chart Object.
Type	UINT
Default	PEDLT_PERCENTAGE
Custom	PEP_nCDATALABELTYPE
VBX	DataLabelType

Purpose

This property controls whether the Pie Chart Object displays slice labels as percentages or actual values.

The following constants apply to this property:

<u>Constant</u>	Description
PEDLT_PERCENTAGE	Data Labels as Percentages.
PEDLT_VALUE	Data Labels as Values.

PEP_szaDATAPOINTLABELS

Scope	Scientific Graph Object.
Type	Property Array of Tab Delimited Strings
Default	empty
Custom	NA
VBX	DataPointLabels

Purpose

This property identifies data labels which will be placed next to data points when PEP_nALLOWDATALABELS equals PEADP_DATAPOINTLABELS and PEP_bGRAPHDATALABELS equals TRUE. Each string in the tab delimited array should be less than or equal to 48 characters in length.

Comments

When used with PEvget, the destination buffer must be large enough to hold all of the tab delimited string information. For property arrays, calling PEvget with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

C / C++ Example

```
/* set 4 tab delimited data point labels */
char szDataPointLabels[] = Texas\tCalifornia\tFlorida\tWashington\t;
PEvset (hWndPE, PEP_szaDATAPOINTLABELS, szDataPointLabels, 4);
```

VBX Example

```
* array bounds are from 0 to (Subsets*Points)-1 *
PEGraph1.DataPointLabels(0) = Texas
PEGraph1.DataPointLabels(1) = California
PEGraph1.DataPointLabels(2) = Florida
PEGraph1.DataPointLabels(3) = Washington
```

PEP_nDATAPRECISION

Scope	All ProEssentials Objects.
Type	UINT
Default	3
Custom	PEP_nCDATAPRECISION
VBX	DataPrecision

Purpose

This property controls the number of decimal positions that will be used in outputting data to object Tables, DataLabels, and the Clipboard. Possible values are:

<u>Value</u>	Description
0	No decimal positions.
1	1 decimal position
2	2 decimal positions
3	3 decimal positions

PEP_bDATASHADOWS

Scope	All ProEssentials Objects.
Type	BOOL
Default	TRUE
Custom	PEP_bCDATASHADOWS
VBX	DataShadows

Purpose

This property controls whether shadows will be placed behind plotting method graphics. The shadows add depth to the images, however, they will slow image creation significantly when graphing many subsets and/or points.

Setting	Description
TRUE	Draw shadows.
FALSE	Do not draw shadows.

PEP_nDEFORIENTATION

Scope	All ProEssentials Objects.
Type	UINT
Default	PEDO_LANDSCAPE
Custom	NA
VBX	DefOrientation

Purpose

This property controls the default orientation of the printer paper when printing a ProEssentials object.

Possible values are as follows:

<u>Constant</u>	Description
PEDO_DRIVERDEFAULT	Use the printer drivers default orientation.
PEDO_LANDSCAPE	Landscape orientation.
PEDO_PORTRAIT	Portrait orientation.

PEP_dwDESKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCDESKCOLOR
VBX	DeskColor

Purpose

This property defines the color used as the background color on which the control is placed.

Comments

PEP_dwTEXTCOLOR is used as the foreground color.

PEP_bDIALOGSHOWN

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NA

Purpose

This property represents a flag that identifies whether the object has a modal dialog currently invoked.

Comments

If you invoke a developer defined modal dialog which is a child of the ProEssentials object, set this property to TRUE before invoking the modal dialog, and reset this property to FALSE when the modal dialog returns.

PEP_bDIRTY

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NA

Purpose

This property is a flag that identifies whether the user has adjusted any properties and or size of the object. The OLE server, PEGRPSVR, uses this property to determine if the user should be prompted to save any adjustments made to the object before closing the server.

PEP_nFONTSIZE

Scope	All ProEssentials Objects.
Type	UINT
Default	PEFS_MEDIUM
Custom	PEP_nCFONTSIZE
VBX	FontSize

Purpose

This property controls the fontsize used in the image creation process. The font size is identified as Large, Medium, or Small via the following constants:

Constant	Description
PEFS_LARGE	Large Font
PEFS_MEDIUM	Medium Font
PEFS_SMALL	Small Font

PEP_bFORCERIGHTYAXIS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	ForceRightYAxis

Purpose

This property controls whether a right Y axis should always be displayed along with the left Y axis.

Setting	Description
TRUE	Force right Y axis.
FALSE	Do not force right Y axis.

Comments

When setting this property to TRUE, the left and right Y axis extents should be manually set via PEP_MANUALSCALECONTROLX and PEP_nMANUALSCALECONTROLRY.

PEP_bFORCEVERTICALPOINTS

Scope	Graph Object.
Type	BOOL
Default	FALSE
Custom	PEP_bCFORCEVERTICALPOINTS
VBX	ForceVerticalPoints

Purpose

This property controls whether point labels are forced into vertical orientation or the orientation is automatically determined by the object.

Setting	Description
TRUE	Force into vertical orientation.
FALSE	Automatically determine orientation.

PEP_dwGRAPHBACKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCGRAPHBACKCOLOR
VBX	GraphBackColor

Purpose

This property identifies the color used as the background color for the objects graphing area.

Comments

PEP_dwGRAPHFORECOLOR is the respective foreground color.

PEP_bGRAPHDATALABELS

Scope	Scientific Graph Object.
Type	BOOL
Default	FALSE
Custom	PEP_bCGRAPHDATALABELS
VBX	GraphDataLabels

Purpose

This property controls whether data labels are placed next to data points in the objects plotting method.

Setting	Description
TRUE	Place data labels.
FALSE	Do not place data labels..

PEP_dwGRAPHFORECOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	BLACK RGB(0,0,0)
Custom	PEP_dwCGRAPHFORECOLOR
VBX	GraphForeColor

Purpose

This property identifies the color used as the foreground color for the objects graphing area.

Comments

PEP_dwGRAPHBACKCOLOR is the respective background color.

PEP_nGRAPHPLUSTABLE

Scope	Graph Object.
Type	UINT
Default	PEGPT_GRAPH
Custom	PEP_nCGRAPHPLUSTABLE
VBX	GraphPlusTable

Purpose

This property controls whether the Graph Object displays a graph, table, or both a graph and table.

Possible values are as follows:

<u>Constant</u>	Description
PEGPT_GRAPH	Display a graph.
PEGPT_TABLE	Display a table.
PEGPT_BOTH	Display a graph and table.

PEP_bGRIDINFRONT

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCGRIDINFRONT
VBX	GridInFront

Purpose

This property controls whether the graphs grid is placed behind or in front of the plotting method graphics.

Setting

TRUE
FALSE

Description

Draw grid after plotting method.
Draw grid before plotting method.

PEP_nGRIDLINECONTROL

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEGLC_BOTH
Custom	PEP_nCGRIDLINECONTROL
VBX	GridLineControl

Purpose

This property controls the horizontal and vertical lines which make the graphs grid.

Possible values are as follows:

<u>Constant</u>	Description
PEGLC_BOTH	Both horizontal and vertical grid lines.
PEGLC_YAXIS	Horizontal grid lines.
PEGLC_XAXIS	Vertical grid lines.
PEGLC_NONE	No grid lines.

PEP_nGROUPINGPERCENT

Scope	Pie Chart Object.
Type	UINT
Default	3
Custom	PEP_nCGROUPINGPERCENT
VBX	GroupingPercent

Purpose

This property controls the size of Pie Chart slices that get grouped into the *Other slice. The *Other slice is used to make the graph more readable by pooling all the small sliver type pie slices into one larger pie slice.

Possible values are as follows:

<u>Value</u>	Description
0	No pie slices are grouped.
1	Slices 1% and smaller are grouped.
2	Slices 2% and smaller are grouped.
3	Slices 3% and smaller are grouped.
4	Slices 4% and smaller are grouped.
5	Slices 5% and smaller are grouped.

PEP_structHOTSPOTDATA

Scope	All ProEssentials Objects.
Type	structure
Default	empty
Custom	NA
VBX	NA

Purpose

This read-only property sends a HOTSPOTDATA structure describing the most recent Hot-Spot event.

Comments

This property is only used when one of the following Hot-Spots are enabled:

PEP_bALLOWSUBSETHOTSPOTS, PEP_bALLOWPOINTHOTSPOTS,
PEP_bALLOWDATAHOTSPOTS, PEP_bALLOWGRAPHHOTSPOTS,
PEP_bALLOWTABLEHOTSPOTS.

In a SDK implementation which supports Hot-Spots, the developers should use PEvget in response to a PEWN_CLICKED, or PEWN_DBLCLICKED notification messages. Processing the data passed back from PEvget allows the developer to determine if a Hot-Spot was clicked, and of what type.

In a VBX implementation this property is optional. The ProEssentials VBX interface will automatically decipher the contents of this property and fire one of the respective VBX events: SubsetHotSpot, PointHotSpot, DataHotSpot, GraphHotSpot, and TableHotSpot.

PEP_nINITIALSCALEFORRYDATA

Scope	Graph and Scientific Graph Objects.
Type	INT
Default	0
Custom	NA
VBX	NA

Purpose

This property is used to tell the object that the right Y axis comparison subset data passed into the object via the PEP_faYDATA property is actually scaled by 10 to the PEP_nINITIALSCALEFORRYDATA power.

Comments

This property can be used to overcome the size limitation (3.4E +/- 38) of the four-byte floating point data managed by the object.

During the PEreinitialize function the object can automatically scale the data as needed. To determine PEP_faYDATA's true value, multiply the data by 10 to the PEP_nSCALEFORRYDATA power.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_nINITIALSCALEFORXDATA

Scope	Scientific Graph Object.
Type	INT
Default	0
Custom	NA
VBX	NA

Purpose

This property is used to tell the object that the data passed into the object via the PEP_faXDATA property is actually scaled by 10 to the PEP_nINITIALSCALEFORXDATA power.

Comments

This property can be used to overcome the size limitation (3.4E +/- 38) of the four-byte floating point data managed by the object.

During the PEreinitialize function the object can automatically scale the data as needed. To determine PEP_faXDATA's true value, multiply the data by 10 to the PEP_nSCALEFORXDATA power.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_nINITIALSCALEFORYDATA

Scope	Graph and Scientific Graph Objects.
Type	INT
Default	0
Custom	NA
VBX	NA

Purpose

This property is used to tell the object that the data passed into the object via the PEP_faYDATA property is actually scaled by 10 to the PEP_nINITIALSCALEFORYDATA power.

Comments

This property can be used to overcome the size limitation (3.4E +/- 38) of the four-byte floating point data managed by the object.

During the PEreinitialize function the object can automatically scale the data as needed. To determine PEP_faYDATA's true value, multiply the data by 10 to the PEP_nSCALEFORYDATA power.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_bLABELBOLD

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCLABELBOLD
VBX	LabelBold

Purpose

This property determines if the labels should be bold.

Setting	Description
TRUE	Bold labels.
FALSE	Normal labels.

Comments

Labels include the Axis Labels, Subset Labels, Point Labels, and Grid Numbers.

PEP_szLABELFONT

Scope	All ProEssentials Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Arial
Custom	PEP_szCLABELFONT
VBX	LabelFont

Purpose

This property identifies the facename of the font to be used for object labels.

Comments

Only assign True-Type fonts to this property.

Labels include the Axis Labels, Subset Labels, Point Labels, and Grid Numbers. This is the same name that is used in the Windows **CreateFont** function.

PEP_bLABELITALIC

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCLABELITALIC
VBX	LabelItalic

Purpose

This property determines if the labels should be italic.

Setting	Description
TRUE	Italic labels.
FALSE	Normal labels.

Comments

Labels include the Axis Labels, Subset Labels, Point Labels, and Grid Numbers.

PEP_bLABELUNDERLINE

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCLABELUNDERLINE
VBX	LabelUnderline

Purpose

This property determines if the labels should be underlined.

Setting	Description
TRUE	Underline labels.
FALSE	Normal labels.

Comments

Labels include the Axis Labels, Subset Labels, Point Labels, and Grid Numbers.

PEP_rectLOGICALLOC

Scope	All ProEssentials Objects.
Type	RECT structure
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to provide auto-sizing capabilities to the object. This RECT defines the logical location of the object with respect to the PEP_nPAGEWIDTH, and PEP_nPAGEHEIGHT logical extents of the parent windows client area. These extents are developer defined.

Comments

This property is used in conjunction with the PEP_nPAGEWIDTH and PEP_nPAGEHEIGHT properties, and PEWM_PARENTVIEW_RESIZED message.

When the parent window is resized, it should send the PEWM_PARENTVIEW_RESIZED message to all ProEssentials controls that it contains. This will cause the ProEssentials controls to resize themselves with respect to their parent window.

For more information see the description for the PEWM_PARENTVIEW_RESIZED message in Appendix C.

PEP_szLOWERBOUNDTEXT

Scope	Graph and Scientific Graph Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Lower
Custom	NA
VBX	LowerBoundText

Purpose

This property identifies the text used to label the lower boundary. Boundaries are upper and/or lower graph sections that identify important data values.

Comments

This property is used in conjunction with PEP_fLOWERBOUNDVALUE, PEP_fUPPERBOUNDVALUE, PEP_szUPPERBOUNDTEXT, and PEP_nVBOUNDARYTYPES.

PEP_fLOWERBOUNDVALUE

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	0.0F
Custom	NA
VBX	LowerBoundValue

Purpose

This property identifies the value that is used to control placement of the lower boundary. Boundaries are upper and/or lower graph sections that identify important data values.

Comments

This property is used in conjunction with PEP_szLOWERBOUNDTEXT, PEP_fUPPERBOUNDVALUE, PEP_szUPPERBOUNDTEXT, and PEP_nVBOUNDARYTYPES.

If the values defined by PEP_fLOWERBOUNDVALUE is not close to the data range of the objects data, then the boundary will not be shown.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_szMAINTITLE

Scope	All ProEssentials Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Main Title
Custom	NA
VBX	MainTitle

Purpose

This property identifies the Main Title of the object. This title is the top most title and centered inside the objects bounding rect.

PEP_bMAINTITLEBOLD

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCMAINTITLEBOLD
VBX	MainTitleBold

Purpose

This property determines if the Main Title should be bold.

Setting	Description
TRUE	Bold Main Title.
FALSE	Normal Main Title.

PEP_szMAINTITLEFONT

Scope	All ProEssentials Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Times New Roman
Custom	PEP_szCMAINTITLEFONT
VBX	MainTitleFont

Purpose

This property identifies the facename of the font used on the Main Title of the object.

Comments

Only assign True-Type fonts to this property.

This is the same name that is used in the Windows **CreateFont** function.

PEP_bMAINTITLEITALIC

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bcMAINTITLEITALIC
VBX	MainTitleItalic

Purpose

This property determines if the Main Title should be italic.

Setting	Description
TRUE	Italic Main Title.
FALSE	Normal Main Title.

PEP_bMAINTITLEUNDERLINE

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCMAINTITLEUNDERLINE
VBX	MainTitleUnderline

Purpose

This property determines if the Main Title should be underlined.

Setting

TRUE
FALSE

Description

Underline Main Title.
Normal Main Title.

PEP_szMANUALMAXDATASTRING

Scope	Graph Object.
Type	NULL Terminated String <= 48 characters in length.
Default	empty
Custom	NA
VBX	ManualMaxDataString

Purpose

This property is **only** used in a real-time implementation. This property represents a data value which will be used to calculate the maximum data string length which will be placed into the Graph Objects table.

Comments

If this value is not set for a real-time implementation, the Graph Object could overlap text or waste space in the table portion of the Graph Objects image.

Similarly, PEP_szMANUALMAXPOINTLABEL should also be set in a real-time implementation.

PEP_fMANUALMAXRY

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	100.0
Custom	NA
VBX	ManualMaxRY

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROLRY. If PEP_nMANUALSCALECONTROLRY is equal to PEMSC_MAX or PEMSC_MINMAX, then the value defined by PEP_fMANUALMAXRY will be the maximum extent of the right \bar{Y} axis grid.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_fMANUALMAXX

Scope	Scientific Graph Object.
Type	FLOAT
Default	100.0
Custom	NA
VBX	ManualMaxX

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROLX. If PEP_nMANUALSCALECONTROLX is equal to PEMSC_MAX or PEMSC_MINMAX, then the value defined by PEP_fMANUALMAXX will be considered in determining the range of PEP_faXDATA.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_fMANUALMAXY

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	100.0
Custom	NA
VBX	ManualMaxY

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROL. If PEP_nMANUALSCALECONTROL is equal to PEMSC_MAX or PEMSC_MINMAX, then the value defined by PEP_fMANUALMAXY will be considered in determining the range of PEP_faYDATA.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_szMANUALMAXPOINTLABEL

Scope	Graph Object.
Type	NULL Terminated String <= 48 characters in length.
Default	empty
Custom	NA
VBX	ManualMaxPointLabel

Purpose

This property is **only** used in a real-time implementation. This property represents a point label which will be used to calculate the maximum point label length in determining the size of area for point label placement.

Comments

If this value is not set for a real-time implementation, the Graph Object could overlap text or waste space in the point label portion of the Graph Objects image.

Similarly, PEP_szMANUALMAXDATASTRING should also be set in a real-time implementation.

PEP_fMANUALMINRY

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	0.0
Custom	NA
VBX	ManualMinRY

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROLRY. If PEP_nMANUALSCALECONTROLRY is equal to PEMSC_MIN or PEMSC_MINMAX, then the value defined by PEP_fMANUALMINRY will be the minimum extent of the right Y axis grid.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_fMANUALMINX

Scope	Scientific Graph Object.
Type	FLOAT
Default	0.0
Custom	NA
VBX	ManualMinX

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROLX. If PEP_nMANUALSCALECONTROLX is equal to PEMSC_MIN or PEMSC_MINMAX, then the value defined by PEP_fMANUALMINX will be considered in determining the range of PEP_faXDATA.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_fMANUALMINY

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	0.0
Custom	NA
VBX	ManualMinY

Purpose

This property is used in conjunction with PEP_nMANUALSCALECONTROL. If PEP_nMANUALSCALECONTROL is equal to PEMSC_MIN or PEMSC_MINMAX, then the value defined by PEP_fMANUALMINY will be considered in determining the range of PEP_faYDATA.

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_fMANUALRYAXISLINE

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualRYAxisLine

Purpose

This property provides manual control of the density of grid lines. Grid lines will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALRYAXISTICKNLINE and PEP_fMANUALRYAXISTICK.

It is important that the value of this property be divisible by the value of PEP_fMANUALRYAXISTICK. The Graph Object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If the right Y axis extents go from 0 to 50 and this property is equal to 10, there will be grid lines at 10, 20, 30, and 40.

PEP_fMANUALRYAXISTICK

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualRYAxisTick

Purpose

This property provides manual control of the density of grid tick marks. Grid tick marks will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALRYAXISTICKNLINE and PEP_fMANUALRYAXISLINE.

It is important that the value PEP_fMANUALRYAXISLINE be divisible by this value. The Graph Object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If right Y Axis extents are from 0 to 50, PEP_fMANUALRYAXISLINE is equal to 10, and this property is equal to 2, there will be grid tick marks at 2, 4, 6, 8, 12, 14, 16, 18, 22, 24, 26, 28...

PEP_bMANUALRYAXISTICKNLIN

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	ManualRYAxisTicknLine

Purpose

This property determines whether the right Y axis has manual grid line and tick mark densities.

Setting

TRUE
FALSE

Description

Manual grid line and tick mark densities.
Automatic grid line and tick mark densities.

Comments

This property is used in conjunction with PEP_fMANUALRYAXISTICK and PEP_fMANUALRYAXISLINE.

PEP_nMANUALSCALECONTROLRY

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEMSC_NONE
Custom	NA
VBX	ManualScaleControlRY

Purpose

This property is used in conjunction with PEP_fMANUALMINRY and PEP_fMANUALMAXRY. This property allows manual control of the range of the graphs right Y axis scale. Possible values are:

Constant	Description
PEMSC_NONE	No Manual Control
PEMSC_MIN	Minimum Scale Control
PEMSC_MAX	Maximum Scale Control
PEMSC_MINMAX	Both Minimum and Maximum Scale Control

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_nMANUALSCALECONTROLX

Scope	Scientific Graph Object.
Type	UINT
Default	PEMSC_NONE
Custom	NA
VBX	ManualScaleControlX

Purpose

This property is used in conjunction with PEP_fMANUALMINX and PEP_fMANUALMAXX. This property allows manual control of the range of the graphs scale. Possible values are:

Constant	Description
PEMSC_NONE	No Manual Control
PEMSC_MIN	Minimum Scale Control
PEMSC_MAX	Maximum Scale Control
PEMSC_MINMAX	Both Minimum and Maximum Scale Control

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_nMANUALSCALECONTROL

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEMSC_NONE
Custom	NA
VBX	ManualScaleControlY

Purpose

This property is used in conjunction with PEP_fMANUALMINY and PEP_fMANUALMAXY. This property allows manual control of the range of the graphs scale. Possible values are:

Constant	Description
PEMSC_NONE	No Manual Control
PEMSC_MIN	Minimum Scale Control
PEMSC_MAX	Maximum Scale Control
PEMSC_MINMAX	Both Minimum and Maximum Scale Control

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call PEreinitialize to complete the objects transformation.

PEP_fMANUALSTACKEDMAXY

Scope	Graph Object.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualStackedMaxY

Purpose

This property provides a manual maximum for stacked style plotting methods.

Comments

This property is only used when PEP_nMANUALSCALECONTROL is non-zero and the value of this property is greater than zero.

PEP_fMANUALXAXISLINE

Scope	Scientific Graph Object.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualXAxisLine

Purpose

This property provides manual control of the density of grid lines. Grid lines will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALXAXISTICKNLINE and PEP_fMANUALXAXISTICK.

It is important that the value of this property be divisible by the value of PEP_fMANUALXAXISTICK. The object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If the X axis extents go from 0 to 50 and this property is equal to 10, there will be grid lines at 10, 20, 30, and 40.

PEP_fMANUALXAXISTICK

Scope	Scientific Graph Object.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualXAxisTick

Purpose

This property provides manual control of the density of grid tick marks. Grid tick marks will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALXAXISTICKNLIN and PEP_fMANUALXAXISLINE.

It is important that the value PEP_fMANUALXAXISLINE be divisible by this value. The object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If X Axis extents are from 0 to 50, PEP_fMANUALXAXISLINE is equal to 10, and this property is equal to 2, there will be grid tick marks at 2, 4, 6, 8, 12, 14, 16, 18, 22, 24, 26, 28...

PEP_bMANUALXAXISTICKNLINE

Scope	Scientific Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	ManualXAxisTicknLine

Purpose

This property determines whether the X axis has manual grid line and tick mark densities.

Setting

TRUE
FALSE

Description

Manual grid line and tick mark densities.
Automatic grid line and tick mark densities.

Comments

This property is used in conjunction with PEP_fMANUALXAXISTICK and PEP_fMANUALXAXISLINE.

PEP_fMANUALYAXISLINE

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualYAxisLine

Purpose

This property provides manual control of the density of grid lines. Grid lines will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALYAXISTICKNLINE and PEP_fMANUALYAXISTICK.

It is important that the value of this property be divisible by the value of PEP_fMANUALYAXISTICK. The object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If the Y axis extents go from 0 to 50 and this property is equal to 10, there will be grid lines at 10, 20, 30, and 40.

PEP_fMANUALYAXISTICK

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	1.0
Custom	NA
VBX	ManualYAxisTick

Purpose

This property provides manual control of the density of grid tick marks. Grid tick marks will be placed at multiples of the defined value.

Comments

This property is used in conjunction with PEP_bMANUALYAXISTICKNLINE and PEP_fMANUALYAXISLINE.

It is important that the value PEP_fMANUALYAXISLINE be divisible by this value. The object will override manual grid line and tick parameters if tick or line densities are too small.

Also, if the user is going to have access to changing the shape and or size of the object, it is suggested that setting manual tick and line densities be avoided.

Example

If Y Axis extents are from 0 to 50, PEP_fMANUALYAXISLINE is equal to 10, and this property is equal to 2, there will be grid tick marks at 2, 4, 6, 8, 12, 14, 16, 18, 22, 24, 26, 28...

PEP_bMANUALYAXISTICKNLIN

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	ManualYAxisTicknLine

Purpose

This property determines whether the Y axis has manual grid line and tick mark densities.

Setting

TRUE
FALSE

Description

Manual grid line and tick mark densities.
Automatic grid line and tick mark densities.

Comments

This property is used in conjunction with PEP_fMANUALYAXISTICK and PEP_fMANUALYAXISLINE.

PEP_bMARKDATAPOINTS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCMARKDATAPOINTS
VBX	MarkDataPoints

Purpose

This property controls whether data-points are marked with a small circle.

Setting	Description
TRUE	Mark data-points.
FALSE	Do not mark data-points.

PEP_nMAXPOINTSTOGRAPH

Scope	Graph Object.
Type	UINT
Default	100
Custom	NA
VBX	MaxPointsToGraph

Purpose

This property controls the maximum number of points which can be graphed. The Customization Dialog will limit the Points To Graph scrollbar to a maximum set by this property.

PEP_hMEMBITMAP

Scope	All ProEssentials Objects.
Type	HANDLE
Default	NULL
Custom	NA
VBX	NA

Purpose

This property is an advanced property. This property identifies a display compatible memory bitmap that is used to prepare the objects image in memory.

Comments

This property is Read Only.

The memory bitmap is only valid if PEP_bPREPAREIMAGES equals TRUE. The image in the bitmap is up-to-date after the object gets the WM_PAINT message.

PEP_hMEMDC

Scope	All ProEssentials Objects.
Type	HANDLE
Default	NULL
Custom	NA
VBX	NA

Purpose

This property is an advanced property. This property identifies a handle to a memory device context that is used to prepare the objects image in memory.

Comments

This property is Read Only.

The memory DC is only valid if PEP_bPREPAREIMAGES equals TRUE. The image in the DC is up-to-date after the object gets the WM_PAINT message.

PEP_dwMONODESKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCMONODESKCOLOR
VBX	NA

Purpose

This property identifies the monochrome-version background color on which the object is placed.

PEP_dwMONOGRAPHBACKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCMONOGRAPHBACKCOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the background color for the objects graphing area.

PEP_dwMONOGRAPHFORECOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCMONOGRAPHFORECOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the foreground color for the objects graphing area.

PEP_dwMONOSHADOWCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCMONOSHADOWCOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the shadow color placed behind the objects graph and table areas.

PEP_dwMONOTABLEBACKCOLOR

Scope	Graph Object.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCMONOTABLEBACKCOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the background color for the Graph Objects table.

PEP_dwMONOTABLEFORECOLOR

Scope	Graph Object.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCMONOTABLEFORECOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the foreground color for the Graph Objects table.

PEP_dwMONOTEXTCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCMONOTEXTCOLOR
VBX	NA

Purpose

This property identifies the monochrome-version color used as the text color for the objects titles and labels.

PEP_bMONOWITHSYMBOLS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	MonoWithSymbols

Purpose

This property determines whether the monochrome viewing style uses shades or subset symbols.

Setting	Description
TRUE	Monochrome style uses symbols.
FALSE	Monochrome style uses shades.

PEP_bNEGATIVEFROMXAXIS

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	TRUE
Custom	NA
VBX	NegativeFromXAxis

Purpose

This property controls how Bar, Area, and Stick plotting methods handle negative data-points.

Setting

TRUE
FALSE

Description

Draw negative data-points from X Axis.
Draw negative data-points from bottom of graph.

PEP_bNORANDOMPOINTSTOGRAPH

Scope	Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NoRandomPointsToGraph

Purpose

This property controls whether the user will have access to random point selection in the Points to Graph section of the Customization Dialog.

Setting	Description
TRUE	Disable random points to graph.
FALSE	Enable random points to graph.

Comments

If the object is going to have many points, it is probably good to disable random points to graph capabilities. This will allow the Customization Dialog to load quicker.

PEP_bNOSCROLLINGSUBSETCONTROL

Scope	Graph and Scientific Graph Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NoScrollingSubsetControl

Purpose

This property controls whether the user will have access to adjusting Subsets To Graph and/or Scrolling Subset functionality.

Setting

Description

TRUE	Disable Subsets To Graph controls.
FALSE	Enable Subsets To Graph controls.

Comments

If the object is going to serve as a scrollbar to control other related subset information, then this property will allow you to prevent the user from adjusting the Subsets To Graph and/or the number of Scrolling Subsets. When the object is maximized, this property is disregarded.

PEP_bNOSTACKEDDATA

Scope	Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	NoStackedData

Purpose

This property controls whether the Graph Object will support the stacked-data plotting methods. These plotting methods include: Area Stacked, Area Stacked Percent, Bar Stacked, and Bar Stacked Percent.

Setting	Description
TRUE	Disable stacked plotting methods.
FALSE	Enable stacked plotting methods.

Comments

If the Graph Object is going to maintain a large number of subsets, (>14) then the stacked style plotting methods should be disabled.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_fNULLDATAVALUE

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	0.0
Custom	NA
VBX	NullDataValue

Purpose

This property defines the value which is designated as NULL. Plotting method graphics, tables, and mathematical calculations will ignore NULL data values.

PEP_nOBJECTTYPE

Scope	All ProEssentials Objects.
Type	UINT
Default	NA
Custom	NA
VBX	ObjectType

Purpose

This read only property identifies the object type.

Constant

PECONTROL_GRAPH
PECONTROL_SGRAPH
PECONTROL_PIE

Description

Graph Object.
Scientific Graph Object.
Pie Chart Object.

PEP_nPAGEHEIGHT

Scope	All ProEssentials Objects.
Type	INT
Default	0
Custom	NA
VBX	NA

Purpose

This property represents the logical height of the objects parent client area.

Comments

This property is used in conjunction with the PEP_nPAGEWIDTH, PEP_rectLOGICALLOC properties and the PEWM_PARENTVIEW_RESIZED message.

In response to the PEWM_PARENTVIEW_RESIZED message, the object will automatically reposition itself using its logical coordinates identified by PEP_rectLOGICALLOC, within the logical coordinates of the parents client area defined by PEP_nPAGEHEIGHT and PEP_nPAGEWIDTH.

For more information see the description for the PEWM_PARENTVIEW_RESIZED message in Appendix C.

PEP_nPAGEWIDTH

Scope	All ProEssentials Objects.
Type	INT
Default	0
Custom	NA
VBX	NA

Purpose

This property represents the logical width of the objects parent client area.

Comments

This property is used in conjunction with the PEP_nPAGEHEIGHT, PEP_rectLOGICALLOC properties and the PEWM_PARENTVIEW_RESIZED message.

In response to the PEWM_PARENTVIEW_RESIZED message, the object will automatically reposition itself using its logical coordinates identified by PEP_rectLOGICALLOC, within the logical coordinates of the parents client area defined by PEP_nPAGEHEIGHT and PEP_nPAGEWIDTH.

For more information see the description for the PEWM_PARENTVIEW_RESIZED message in Appendix C.

PEP_nPLOTINGMETHOD

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	0
Custom	PEP_nCPLOTINGMETHOD
VBX	PlottingMethod

Purpose

This property determines the plotting method used to render the objects data-set. Depending on the object type, the following constants apply:

Graph Plotting Methods

PEGPM_LINE
PEGPM_BAR
PEGPM_POINT
PEGPM_AREA
PEGPM_AREASTACKED
PEGPM_AREASTACKEDPERCENT
PEGPM_BARSTACKED
PEGPM_BARSTACKEDPERCENT
PEGPM_POINTSPLUSBFL
PEGPM_POINTSPLUSBFLGRAPHED
PEGPM_POINTPLUSBFC
PEGPM_POINTPLUSBFCGRAPHED
PEGPM_POINTSPLUSLINE
PEGPM_POINTPLUSSPLINE
PEGPM_SPLINE
PEGPM_HISTOGRAM
PEGPM_SPECIFICPLOTMODE

There are two versions of Points plus Best Fit Line and Points plus Best Fit Curve. One version calculates the Best Fit with respect to all point information. The other version calculates the Best Fit for only those points which are currently graphed.

Scientific Graph Plotting Methods

PEGPM_LINE
PEGPM_POINT
PEGPM_STICK
PEGPM_POINTSPLUSBFL
PEGPM_POINTSPLUSBFC
PEGPM_POINTSPLUSLINE
PEGPM_POINTPLUSSPLINE
PEGPM_SPLINE
PEGPM_BUBBLE

PEP_szaPOINTLABELS

Scope	Graph and Pie Chart Objects.
Type	Property Array of Tab Delimited Strings
Default	empty
Custom	NA
VBX	PointLabels

Purpose

This property identifies point labels. Each string in the tab delimited array should be less than or equal to 48 characters in length.

Comments

When used with **PEvget**, the destination buffer must be large enough to hold all of the tab delimited string information. For property arrays, calling **PEvget** with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

C / C++ Example

```
/* set 4 tab delimited point labels */  
char szPointLabels[] = Texas\tCalifornia\tFlorida\tWashington\t;  
PEvset (hWndPE, PEP_szaPOINTLABELS, szPointlabels, 4);
```

VBX Example

```
* the Points property must be equal to 4 *  
PEGraph1.PointLabels(0) = Texas  
PEGraph1.PointLabels(1) = California  
PEGraph1.PointLabels(2) = Florida  
PEGraph1.PointLabels(3) = Washington
```

PEP_nPOINTS

Scope	All ProEssentials Objects.
Type	UINT
Default	4
Custom	NA
VBX	Points

Purpose

This property represents the number of points that the object will contain.

Comments

PEP_nSUBSETS * PEP_nPOINTS equals the total number of data values that the object will contain.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_nPOINTSIZ

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEPS_MEDIUM
Custom	NA
VBX	PointSize

Purpose

This property controls the size of plotting method points. Possible values are as follows:

Constant	Description
PEPS_SMALL	Small points.
PEPS_MEDIUM	Medium points.
PEPS_LARGE	Large points.

PEP_nPOINTSTOGRAPH

Scope	Graph Object.
Type	UINT
Default	0
Custom	PEP_nCPOINTSTOGRAPH
VBX	PointsToGraph

Purpose

This property controls the number of points to be viewed. If this value is Zero or is equal to PEP_nPOINTS, then all of the Graphs points will be displayed. If this value is less than PEP_nPOINTS, then only that quantity of points will be displayed and a horizontal scrollbar will allow access to the other point information.

PEP_nPOINTSTOGRAPHINIT

Scope	Graph Object.
Type	UINT
Default	PEPTGI_FIRSTPOINTS
Custom	NA
VBX	PointsToGraphInit

Purpose

This property controls whether the first points or last points are initially displayed when the object has a horizontal scrollbar. Possible values are:

Constant	Description
PEPTGI_FIRSTPOINTS	First points.
PEPTGI_LASTPOINTS	Last points.

PEP_nPOINTSTOGRAPHVERSION

Scope	Graph Object.
Type	UINT
Default	PEPTGV_SEQUENTIAL
Custom	PEP_nCPOINTSTOGRAPHVERSION
VBX	NA

Purpose

This property controls whether point information is displayed sequentially or randomly. Sequential points represent a subset of point information sized by the PEP_nPOINTSTOGRAPH property. Random points are defined by the property array PEP_naRANDOMPOINTSTOGRAPH.

Possible values are defined by the following constants:

<u>Constant</u>	<u>Description</u>
PEPTGV_SEQUENTIAL	Points are displayed sequentially.
PEPTGV_RANDOM	Random points are displayed.

PEP_bPREPAREIMAGES

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	NA
VBX	PrepareImages

Purpose

This property controls whether the object prepares images in memory, or outputs the image directly to screen.

Setting

TRUE
FALSE

Description

Prepare images in memory.
Output directly to screen.

PEP_naRANDOMPOINTSTOGRAPH

Scope	Graph Object.
Type	Property Array of UINTs defining Zero based indexes.
Default	empty
Custom	PEP_naCRANDOMPOINTSTOGRAPH
VBX	RandomPointsToGraph

Purpose

This property controls which point information is graphed when PEP_nPOINTSTOGRAPHVERSION is equal to PEPTGV_RANDOM.

Comments

At least two point indexes must be supplied.

PEP_naRANDOMSUBSETSTOGRAPH

Scope	Graph and Scientific Graph Objects.
Type	Property Array of UINTs defining Zero based indexes.
Default	empty
Custom	PEP_naCRANDOMSUBSETSTOGRAPH
VBX	RandomSubsetsToGraph

Purpose

This property controls which subsets are displayed. It is dependent on the PEP_nSCROLLINGSUBSETS property.

ScrollingSubsets

Zero

Non-Zero

Purpose

Display only those subset indexes specified.

Make specified subset indexes permanent.

The following examples shows how to specify subset indexes to this property.

C / C++ Example

```
int nArray[2];
nArray[0] = 0;
nArray[1] = 1;
PEvset (hWndPE, PEP_naRANDOMSUBSETSTOGRAPH, nArray, 2)
```

VBX Example

```
PEGraph1.RandomSubsetsToGraph(0) = 0
PEGraph1.RandomSubsetsToGraph(1) = 1
```

PEP_nRYAXISCOMPARISONSUBSETS

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	empty
Custom	NA
VBX	RYAxisComparisonSubsets

Purpose

This property is used to establish custom (developer defined) comparison subsets which are graphed with respect to an alternate right Y Axis.

Comments

This is a pre-initialization property. During the **PEreinitialize** function, the object will use this property to designate the last PEP_nRYAXISCOMPARISONSUBSETS subset indexes as comparison data and then sets this property to zero.

Custom comparison subsets are always located after non-comparison subsets and before auto-stat subsets. Right Y axis comparison subsets are located before left Y axis comparison subsets.

It is the developers responsibility to transfer the comparison data (along with non-comparison data) into the object via the PEP_faXDATA, PEP_faYDATA, and PEP_szaSUBSETLABELS properties, and precede the respective subset labels with an asterisks (which identifies the subset as a comparison subset.)

PEP_szRYAXISLABEL

Scope	Graph and Scientific Graph Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Right Y Axis
Custom	NA
VBX	RYAxisLabel

Purpose

This property controls the text used as the right Y Axis label.

PEP_nRYAXISSCALECONTROL

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEAC_AUTO
Custom	NA
VBX	RYAxisScaleControl

Purpose

This property controls the grid scale used for the right Y axis. Possible values are defined by the following constants:

Constant	Description
PEAC_AUTO	Auto scaling.
PEAC_NORMAL	Normal scale.
PEAC_LOG	Log scale.

Comments

Auto scaling will determine if the object should use a normal or log scale. If the data's range is larger than or equal to 10e3, then a log scale will be used.

If the extents of the scale go to zero or below, the only allowed scale will be PEAC_NORMAL.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_nSCALEFORRYDATA

Scope	Graph and Scientific Graph Objects.
Type	INT
Default	0
Custom	NA
VBX	ScaleForRYData

Purpose

This property is used to determine the right Y axis comparison subset scaling. The objects data can be initially scaled via PEP_nINITIALSCALEFORRYDATA, and further scaled automatically by the object (if PEP_nMANUALSCALECONTROLRY = PEMSC_NONE).

Comments

To determine right Y axis comparison subset true values, multiply the data by 10 to the PEP_nSCALEFORRYDATA power.

This property is Read Only.

PEP_nSCALEFORXDATA

Scope	Scientific Graph Object.
Type	INT
Default	0
Custom	NA
VBX	ScaleForXData

Purpose

This property is used to determine the objects PEP_faXDATA current amount of scaling. The objects data can be initially scaled via PEP_nINITIALSCALEFORXDATA, and further scaled automatically by the object (if PEP_nMANUALSCALECONTROLX = PEMSC_NONE).

Comments

To determine PEP_faXDATA's true value, multiply the data by 10 to the PEP_nSCALEFORXDATA power.

This property is Read Only.

PEP_nSCALEFORYDATA

Scope	Graph and Scientific Graph Objects.
Type	INT
Default	0
Custom	NA
VBX	ScaleForYData

Purpose

This property is used to determine the objects PEP_faYDATA current amount of scaling. The objects data can be initially scaled via PEP_nINITIALSCALEFORYDATA, and further scaled automatically by the object (if PEP_nMANUALSCALECONTROL = PEMSC_NONE).

Comments

To determine PEP_faYDATA's true value, multiply the data by 10 to the PEP_nSCALEFORYDATA power.

This property is Read Only.

PEP_nSCROLLINGSUBSETS

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	0
Custom	PEP_nCSCROLLINGSUBSETS
VBX	ScrollingSubsets

Purpose

This property controls the number of subsets which are revolved via the objects vertical scrollbar.

Comments

The total number of subsets displayed is equal to the number of permanent subsets defined by PEP_naRANDOMSUBSETSTOGRAPH plus the quantity defined by PEP_nSCROLLINGSUBSETS. If this property is Zero then the objects vertical scrollbar is not shown.

PEP_dwSHADOWCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	Gray RGB(128,128,128)
Custom	PEP_dwCSHADOWCOLOR
VBX	ShadowColor

Purpose

This property identifies the color-version color used as the shadow color placed behind the objects graph and table areas.

PEP_nSPECIFICPLOTMODE

Scope	Graph Object.
Type	UINT
Default	PESPM_NONE
Custom	NA
VBX	SpecificPlotMode

Purpose

The Graph Object has complex plotting methods which require multiple subsets. These types of plotting methods are called Specific Plotting Methods. The following constants tell the Graph Object whether a Specific Plotting Method is being graphed and of what type.

Constant	Description
PESPM_NONE	No Specific Plotting Method.
PESPM_HLBAR	Two subsets representing BarStart-BarEnd.
PESPM_HLLINE	Two subsets representing :LineStart-LineEnd.
PESPM_HLC	Three subsets representing LineStart-LineEnd-Close.
PESPM_OHLC	Four subsets representing LineStart-LineEnd-Open-Close.
PESPM_BOX	Four subsets representing LineStart-LineEnd-BoxStart-BoxEnd.

Comments

The subsets should be ordered as described in the above Description section.

The Specific Plotting Method is only displayed when PEP_nPLOTINGMETHOD equals PEGPM_SPECIFICPLOTMODE.

For the Box Plot method you can reverse the order of BoxStart and BoxEnd subsets. This changes the polarity of whether the Box is filled or not-filled.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_dwaSUBSETCOLORS

Scope	All ProEssentials Objects.
Type	Property Array of DWORDs
Default	RGB(255,255,255) RGB(0,255,0) RGB(0,0,255) RGB(255,255,0) RGB(255,0,255) RGB(255,0,0) RGB(0,255,255) RGB(192,192,192) RGB(80,80,80) RGB(128,0,128) RGB(0,128,0) RGB(128,128,128) RGB(0,128,128) RGB(0,0,128) RGB(128,0,0) RGB(128,128,0)
Custom	NA
VBX	NA

Purpose

This property array controls which colors are used to distinguish between multiple subset information in the objects image. If the number of subset colors defined is less than the number of subsets displayed, then the color information is revolved.

Comments

Subset colors are chosen in sequence. For the color-version image, colors equal to PEP_dwGRAPHFORECOLOR or PEP_dwGRAPHBACKCOLOR are skipped.

PEP_dwaSUBSETSHADES is used to control subset color information for the monochrome-version of the image.

PEP_szaSUBSETLABELS

Scope	All ProEssentials Objects.
Type	Property Array of Tab Delimited Strings.
Default	empty
Custom	NA
VBX	NA

Purpose

This property identifies subset labels. Each string in the tab delimited array should be less than or equal to 48 characters in length.

Comments

When used with **PEvget**, the destination buffer must be large enough to hold all of the tab delimited string information. For property arrays, calling **PEvget** with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

C / C++ Example

```
/* set 4 tab delimited subset labels */  
char szSubsets[] = Texas\tCalifornia\tWashington\tFlorida\t;  
PEvset (hWndPE, PEP_szaSUBSETLABELS, szSubsets, 4);
```

VBX Example

```
* the Subsets property must be equal to 4 *  
PEGraph1.SubsetLabels(0) = Texas  
PEGraph1.SubsetLabels(1) = California  
PEGraph1.SubsetLabels(2) = Florida  
PEGraph1.SubsetLabels(3) = Washington
```

PEP_naSUBSETLINETYPES

Scope	Graph and Scientific Objects.
Type	Property Array of UINTs
Default	PELT_THINSOLID
Custom	NA
VBX	SubsetLineTypes

Purpose

This property array controls which line-types are used to distinguish between multiple subset information in the objects image. If the number of line-types defined is less than the number of subsets displayed, then the line-type information is revolved.

Constant	Description
PELT_THINSOLID	Thin Solid Line
PELT_DASH	Dashed Line
PELT_DOT	Dot Line
PELT_DASHDOT	Dash Dot Line
PELT_DASHDOTDOT	Dash Dot Dot
PELT_MEDIUMSOLID	Medium Solid Line
PELT_THICKSOLID	Thick Solid Line

Comments

PEP_naSUBSETPOINTTYPES is used to control subset point-type information.

PEP_naSUBSETPOINTTYPES

Scope	Graph and Scientific Objects.
Type	Property Array of UINTs
Default	PEPT_DOTSOLID
Custom	NA
VBX	SubsetPointTypes

Purpose

This property array controls which point-types are used to distinguish between multiple subset information in the objects image. If the number of point-types defined is less than the number of subsets displayed, then the point-type information is revolved.

Constant	Description
PEPT_PLUS	Plus
PEPT_CROSS	Cross
PEPT_DOT	Circle
PEPT_DOTSOLID	Solid Circle
PEPT_SQUARE	Square
PEPT_SQUARESOLID	Sold Square
PEPT_DIAMOND	Diamond
PEPT_DIAMONDSOLID	Solid Diamond
PEPT_UPTRIANGLE	Upward Triangle
PEPT_UPTRIANGLESOLID	Solid Upward Triangle
PEPT_DOWNTRIANGLE	Downward Triangle
PEPT_DOWNTRIANGLESOLID	Solid Downward Triangle

Comments

If the number of point-types defined is equal or greater than the number of subsets in the object, these same point-types will be used as symbols when PEP_nVIEWINGSTYLE equals PEVS_MONOWITHSYMBOLS.

PEP_naSUBSETLINETYPES is used to control subset line-type information.

PEP_nSUBSETS

Scope	All ProEssentials Objects.
Type	UINT
Default	1
Custom	NA
VBX	Subsets

Purpose

This property represents the number of subsets that the object will contain.

Comments

PEP_nSUBSETS * PEP_nPOINTS equals the total number of data values that the object will contain.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_dwaSUBSETSHADES

Scope	All ProEssentials Objects.
Type	Property Array of DWORDs
Default	RGB(255,255,255) RGB(80,80,80) RGB(192,192,192) RGB(128,128,128)
Custom	NA
VBX	NA

Purpose

For monochrome-version image creation, this property array controls which colors are used to distinguish between multiple subset information in the objects image. If the number of subset shades defined is less than the number of subsets displayed, then the color information is revolved.

Comments

PEP_dwaSUBSETCOLORS is used to control subset color information for the color-version of the image.

PEP_szSUBTITLE

Scope	All ProEssentials Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Sub Title
Custom	NA
VBX	SubTitle

Purpose

This property identifies the objects SubTitle. This title is centered and under the main title if it exists.

PEP_bSUBTITLEBOLD

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCSUBTITLEBOLD
VBX	SubTitleBold

Purpose

This property determines if the SubTitle should be bold.

Setting	Description
TRUE	Bold Sub Title.
FALSE	Normal Sub Title.

PEP_szSUBTITLEFONT

Scope	All ProEssentials Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Times New Roman
Custom	PEP_szCSUBTITLEFONT
VBX	SubTitleFont

Purpose

This property identifies the facename of the font used on the SubTitle of the object.

Comments

Only assign True-Type fonts to this property.

This is the same name that is used in the Windows **CreateFont** function.

PEP_bSUBTITLEITALIC

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCSUBTITLEITALIC
VBX	SubTitleItalic

Purpose

This property determines if the SubTitle should be italic.

Setting	Description
TRUE	Italic SubTitle.
FALSE	Normal SubTitle.

PEP_bSUBTITLEUNDERLINE

Scope	All ProEssentials Objects.
Type	BOOL
Default	FALSE
Custom	PEP_bCSUBTITLEUNDERLINE
VBX	SubTitleUnderline

Purpose

This property determines if the SubTitle should be underlined.

Setting	Description
TRUE	Underline SubTitle.
FALSE	Normal SubTitle.

PEP_nSYMBOLFREQUENCY

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	6
Custom	NA
VBX	SymbolFrequency

Purpose

This property controls the frequency of symbols placed into plotting methods when PEP_nVIEWINGSTYLE is equal to PEVS_MONOWITHSYMBOLS.

<u>Constant</u>	Description
0	Fewest symbols.
...	...
9	Most symbols.

PEP_dwTABLEBACKCOLOR

Scope	Graph Object.
Type	DWORD
Default	Bright White RGB(255,255,255)
Custom	PEP_dwCTABLEBACKCOLOR
VBX	TableBackColor

Purpose

This property identifies the color-version color used as the background color for the Graph Objects table.

PEP_szTABLEFONT

Scope	Graph Object.
Type	NULL Terminated String <= 48 characters in length.
Default	Arial
Custom	PEP_szCTABLEFONT
VBX	TableFont

Purpose

This property identifies the facename of the font used on text inside the Graph Objects table.

Comments

Only assign True-Type fonts to this property.

This is the same name that is used in the Windows **CreateFont** function.

PEP_dwTABLEFORECOLOR

Scope	Graph Object.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCTABLEFORECOLOR
VBX	TableForeColor

Purpose

This property identifies the color-version color used as the foreground color for the Graph Objects table.

PEP_nTABLEWHAT

Scope	Graph Object.
Type	UINT
Default	PETW_GRAPHED
Custom	PEP_nCTABLEWHAT
VBX	TableWhat

Purpose

This property controls what information is included into the objects table. Possible values are defined in the following constants.

Constant	Description
PETW_GRAPHED	Table those subsets which are graphed.
PETW_ALLSUBSETS	Table all subsets,

PEP_nTARGETPOINTSTOTABLE

Scope	Graph Object.
Type	UINT
Default	10
Custom	NA
VBX	TargetPointsToTable

Purpose

This property represents the number of point labels which will be attempted to be displayed when PEP_nPOINTSTOGRAPH is greater than or equal to PEP_nALTFREQTHRESHOLD.

Comments

Also see PEP_naALTFREQUENCIES.

PEP_dwTEXTCOLOR

Scope	ALL ProEssentials Objects.
Type	DWORD
Default	Black RGB(0,0,0)
Custom	PEP_dwCTEXTCOLOR
VBX	TextColor

Purpose

This property identifies the color-version color used as the text color for the objects titles, and labels.

PEP_bTREATCOMPSASNORMAL

Scope	Graph Object.
Type	BOOL
Default	FALSE
Custom	PEP_bCTREATCOMPSASNORMAL
VBX	TreatCompsAsNormal

Purpose

This property controls whether comparison subsets are viewed as comparison/permanent subsets (using a thin line,) or whether they are viewed as normal data (using the current plotting method of the object.)

Setting

TRUE
FALSE

Description

Draw comparisons using current plotting method.
Draw comparisons as thin solid lines.

PEP_szUPPERBOUNDTEXT

Scope	Graph and Scientific Graph Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Upper
Custom	NA
VBX	UpperBoundText

Purpose

This property identifies the text used to label the upper boundary. Boundaries are upper and/or lower graph sections that identify important data values.

Comments

This property is used in conjunction with PEP_fUPPERBOUNDVALUE, PEP_fLOWERBOUNDVALUE, PEP_szLOWERBOUNDTEXT, and PEP_nVBOUNDARYTYPES.

PEP_fUPPERBOUNDVALUE

Scope	Graph and Scientific Graph Objects.
Type	FLOAT
Default	0.0F
Custom	NA
VBX	UpperBoundValue

Purpose

This property identifies the value that is used to control placement of the upper boundary. Boundaries are upper and/or lower graph sections that identify important data values.

Comments

This property is used in conjunction with PEP_szUPPERBOUNDTEXT, PEP_fLOWERBOUNDVALUE, PEP_szLOWERBOUNDTEXT, and PEP_nVBOUNDARYTYPES.

If the values defined by PEP_fUPPERBOUNDVALUE is not close to the data range of the objects data, then the boundary will not be shown.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_nVBOUNDARYTYPES

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEVB_NONE
Custom	NA
VBX	VBoundaryTypes

Purpose

This property controls the placement of boundaries on the grid area of the Graph and Scientific Graph Objects. Possible values are defined by the following constants:

Constant	Description
PEVB_NONE	No boundaries.
PEVB_TOP	Top boundary.
PEVB_BOTTOM	Bottom boundary.
PEVB_TOPANDBOTTOM	Top and Bottom boundaries.

Comments

This property is used in conjunction with PEP_fUPPERBOUNDVALUE, PEP_szUPPERBOUNDTEXT, PEP_fLOWERBOUNDVALUE, and PEP_szLOWERBOUNDTEXT.

If the values defined by PEP_fUPPERBOUNDVALUE and PEP_fLOWERBOUNDVALUE are not close to the data range of the objects data, then the boundaries that are out of range will not be shown.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

PEP_nVIEWINGSTYLE

Scope	All ProEssentials Objects.
Type	UINT
Default	PEVS_COLOR
Custom	PEP_nVIEWINGSTYLE
VBX	ViewingStyle

Purpose

This property controls the viewing style of the object. Possible values are defined by the following constants:

Constant	Description
PEVS_COLOR	Color Style.
PEVS_MONO	Monochrome Style.
PEVS_MONOWITHSYMBOLS	Monochrome with Symbols.

PEP_dwWDESKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) desk color.

Comments

This property is Read Only.

PEP_dwWGRAPHBACKCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) graph background color.

Comments

This property is Read Only.

PEP_dwWGRAPHFORECOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) graph foreground color.

Comments

This property is Read Only.

PEP_dwWSHADOWCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) shadow color.

Comments

This property is Read Only.

PEP_dwWTABLEBACKCOLOR

Scope	Graph and Scientific Graph Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) table background color.

Comments

This property is Read Only.

PEP_dwWTABLEFORECOLOR

Scope	Graph and Scientific Graph Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) table foreground color.

Comments

This property is Read Only.

PEP_dwWTEXTCOLOR

Scope	All ProEssentials Objects.
Type	DWORD
Default	NA
Custom	NA
VBX	NA

Purpose

This property is used to retrieve the working (current) text color.

Comments

This property is Read Only.

PEP_szXAXISLABEL

Scope	Graph and Scientific Graph Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	X Axis
Custom	NA
VBX	XAxisLabel

Purpose

This property controls the text used as the X Axis label.

PEP_nXAXISSCALECONTROL

Scope	Scientific Graph Object.
Type	UINT
Default	PEAC_AUTO
Custom	NA
VBX	XAxisScaleControl

Purpose

This property controls the grid scale used for the X Axis. Possible values are defined by the following constants:

Constant	Description
PEAC_AUTO	Auto scaling.
PEAC_NORMAL	Normal scale.
PEAC_LOG	Log scale.

Comments

Auto scaling will determine if the object should use a normal or log scale. If the data's range is larger than or equal to $10e3$, then a log scale will be used.

If the extents of the scale go to zero or below, the only allowed scale will be PEAC_NORMAL.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_faXDATA

Scope	Scientific Graph and Pie Chart Objects.
Type	Property Array of FLOATs.
Default	empty
Custom	NA
VBX	XData

Purpose

This property is used to transfer data into the object. For i-Subsets (s) and j-Points (p), the data is specified in the following format:

Data Format

s1p1	s1p2	s1p3	s1p4	...	s1pj
s2p1	s2p2	s2p3	s2p4	...	s2pj
s3p1	s3p2	s3p3	s3p4	...	s3pj
...
sip1	sip2	sip3	sip4	...	sipj

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

The Scientific Graph Object uses PEP_faXDATA to identify the data measured against the X Axis of the object.

The Pie Chart Object uses PEP_faXDATA to identify values to be assigned to pie slices. The Pie Chart Object will determine the percentages to be used as slice labels.

When used with **PEvget**, the destination buffer must be large enough to hold all of the information. For property arrays, calling **PEvget** with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

C / C++ Example

```
hWndPE = PEcreate (PECONTROL_SGRAPH, WS_VISIBLE, &rect, hWnd, 1);

PEnset (hWndPE, PEP_nSUBSETS, 4); /* Object has 4 subsets */
PEnset (hWndPE, PEP_nPOINTS, 20); /* Object has 20 points */

float fData[4][20]; /* first place data into fData */

/* set data, 4 subsets * 20 points = 80 elements */
PEvset (hWndPE, PEP_faXDATA, fData, 80);

PEreinitialize (hWndPE);
PEresetimage (hWndPE, 0, 0);
```

VBX Example

```
** For a Scientific Graph Object placed into a form **
```

```
PESGraph1.Subsets = 4
PESGraph1.Points = 20

' ** make some random data **'
For s% = 0 To 3
  For p% = 0 To 19
    o% = ((s% * 20) + p%)
    PESGraph1.XData(o%) = (p% + 1) * 5 + (Rnd * (25))
  Next p%
Next s%

PESGraph1.Refresh
```

PEP_szYAXISLABEL

Scope	Graph and Scientific Graph Objects.
Type	NULL Terminated String <= 48 characters in length.
Default	Y Axis
Custom	NA
VBX	YAxisLabel

Purpose

This property controls the text used as the Y Axis label.

PEP_nYAXISSCALECONTROL

Scope	Graph and Scientific Graph Objects.
Type	UINT
Default	PEAC_AUTO
Custom	NA
VBX	YAxisScaleControl

Purpose

This property controls the grid scale used for the Y Axis. Possible values are defined by the following constants:

Constant	Description
PEAC_AUTO	Auto scaling.
PEAC_NORMAL	Normal scale.
PEAC_LOG	Log scale.

Comments

Auto scaling will determine if the object should use a normal or log scale. If the data's range is larger than or equal to 10e3, then a log scale will be used.

If the extents of the scale go to zero or below, the only allowed scale will be PEAC_NORMAL.

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the object's transformation.

PEP_faYDATA

Scope	All ProEssentials Objects.
Type	Property Array of FLOATs.
Default	empty
Custom	NA
VBX	YData

Purpose

This property is used to transfer data into the object. For i-Subsets (s) and j-Points (p), the data is specified in the following format:

Data Format

s1p1	s1p2	s1p3	s1p4	...	s1pj
s2p1	s2p2	s2p3	s2p4	...	s2pj
s3p1	s3p2	s3p3	s3p4	...	s3pj
...
sip1	sip2	sip3	sip4	...	sipj

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

The Graph and Scientific Graph Objects use PEP_faYDATA to identify the data measured against the Y Axis of the object.

The Pie Chart Object uses PEP_faYDATA to identify default slices to explode. A value of 0 not to explode, and a value of 1 to explode. If the quantity of PEP_faYDATA doesn't equal the amount of **PEP_faXDATA**, then all subsets will be exploded with respect to the first subset of PEP_faYDATA.

When used with **PEvget**, the destination buffer must be large enough to hold all of information. For property arrays, calling **PEvget** with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

C / C++ Example

```
hWndPE = PEcreate (PECONTROL_GRAPH, WS_VISIBLE, &rect, hWnd, 1);

PEnset (hWndPE, PEP_nSUBSETS, 4); /* Object has 4 subsets */
PEnset (hWndPE, PEP_nPOINTS, 20); /* Object has 20 points */

float fData[4][20]; /* first place data into fData */

/* set data, 4 subsets * 20 points = 80 elements */
PEvset (hWndPE, PEP_faYDATA, fData, 80);

PEreinitialize (hWndPE);
PEresetimage (hWndPE, 0, 0);
```

VBX Example

```
** For a Graph Object placed into a form **
```

```
PEGraph1.Subsets = 4
PEGraph1.Points = 20

' ** make some random data **'
For s% = 0 To 3
  For p% = 0 To 19
    o% = ((s% * 20) + p%)
    PEGraph1.YData(o%) = (p% + 1) * 5 + (Rnd * (25))
  Next p%
Next s%

PEGraph1.Refresh
```

PEP_faZDATA

Scope	Scientific Graph Object.
Type	Property Array of FLOATs.
Default	empty
Custom	NA
VBX	ZData

Purpose

The Scientific Graph Object use PEP_faZDATA to control bubble sizes for the Bubble plotting method. The object will evaluate the range of ZData and then proportionally size bubbles with respect to that range.

For i-Subsets (s) and j-Points (p), the data is specified in the following format:

Data Format

s1p1	s1p2	s1p3	s1p4	...	s1pj
s2p1	s2p2	s2p3	s2p4	...	s2pj
s3p1	s3p2	s3p3	s3p4	...	s3pj
...
sip1	sip2	sip3	sip4	...	sipj

Comments

This is a pre-initialization property. After setting this property and other pre-initialization properties, you must call **PEreinitialize** to complete the objects transformation.

When used with **PEvget**, the destination buffer must be large enough to hold all of information. For property arrays, calling **PEvget** with *lpDest* equal to NULL will return the number of bytes needed to hold the data.

PEP_fZOOMMAXX

Scope	Scientific Graph Object.
Type	FLOAT
Default	100.0
Custom	NA
VBX	ZoomMaxX

Purpose

This property controls the maximum X axis extent when PEP_bZOOMMODE is TRUE.

Comments

This property is used in conjunction with PEP_bZOOMMODE, PEP_fZOOMMINX, PEP_fZOOMMAXY, PEP_fZOOMMINY, and PEP_nALLOWZOOMING.

PEP_fZOOMMAXY

Scope	Scientific Graph Object.
Type	FLOAT
Default	100.0
Custom	NA
VBX	ZoomMaxY

Purpose

This property controls the maximum Y axis extent when PEP_bZOOMMODE is TRUE.

Comments

This property is used in conjunction with PEP_bZOOMMODE, PEP_fZOOMMINY, PEP_fZOOMMAXX, PEP_fZOOMMINX, and PEP_nALLOWZOOMING.

PEP_fZOOMMINX

Scope	Scientific Graph Object.
Type	FLOAT
Default	0.0
Custom	NA
VBX	ZoomMinX

Purpose

This property controls the minimum X axis extent when PEP_bZOOMMODE is TRUE.

Comments

This property is used in conjunction with PEP_bZOOMMODE, PEP_fZOOMMAXX, PEP_fZOOMMAXY, PEP_fZOOMMINY, and PEP_nALLOWZOOMING.

PEP_fZOOMMINY

Scope	Scientific Graph Object.
Type	FLOAT
Default	0.0
Custom	NA
VBX	ZoomMinY

Purpose

This property controls the minimum Y axis extent when PEP_bZOOMMODE is TRUE.

Comments

This property is used in conjunction with PEP_bZOOMMODE, PEP_fZOOMMAXY, PEP_fZOOMMAXX, PEP_fZOOMMINX, and PEP_nALLOWZOOMING.

PEP_bZOOMMODE

Scope	Scientific Graph Object.
Type	BOOL
Default	FALSE
Custom	NA
VBX	ZoomMode

Purpose

This property controls whether the Scientific Graph Object is currently zooming. Zooming is the process of using alternate extents for the graphs grid.

Setting	Description
TRUE	Enable Zooming.
FALSE	Disable Zooming.

Comments

This property is used in conjunction with PEP_fZOOMMAXY, PEP_fZOOMMINY, PEP_fZOOMMAXX, PEP_fZOOMMINX, and PEP_nALLOWZOOMING.

The object has a built in mechanism to control zooming extents, however, the developer can manually set zooming extents and enable zooming.

Constants and Enumerations

Allow Data Labels

PEADL_NONE	0
PEADL_DATAVALUES	1
PEADL_POINTLABELS	2
PEADL_DATAPOINTLABELS	3

Allow Zooming

PEAZ_NONE	0
PEAZ_HORIZONTAL	1
PEAZ_VERTICAL	2
PEAZ_HORZANDVERT	3

Auto Explode Pie Slices

PEAE_NONE	0
PEAE_ALLSUBSETS	1
PEAE_INDSUBSETS	2

Auto Stat Subsets

PEAS_SUMPP	51
PEAS_MINAP	1
PEAS_MINPP	52
PEAS_MAXAP	2
PEAS_MAXPP	53
PEAS_AVGAP	3
PEAS_AVGPP	54
PEAS_P1SDAP	4
PEAS_P1SDPP	55
PEAS_P2SDAP	5
PEAS_P2SDPP	56
PEAS_P3SDAP	6
PEAS_P3SDPP	57
PEAS_M1SDAP	7
PEAS_M1SDPP	58
PEAS_M2SDAP	8
PEAS_M2SDPP	59
PEAS_M3SDAP	9
PEAS_M3SDPP	60

Axis Control

PEAC_AUTO	0
PEAC_NORMAL	1
PEAC_LOG	2

Best Fit Degree

PEBFD_2ND	0
PEBFD_3RD	1
PEBFD_4TH	2

Bubble Size

PEBS_SMALL	0
PEBS_MEDIUM	1
PEBS_LARGE	2

Curve Granularity	
PECG_COARSE	0
PECG_MEDIUM	1
PECG_FINE	2
Data Label Type	
PEDLT_PERCENTAGE	0
PEDLT_VALUE	1
Default Orientation	
PEDO_DRIVERDEFAULT	0
PEDO_LANDSCAPE	1
PEDO_PORTRAIT	2
Font Size	
PEFS_LARGE	0
PEFS_MEDIUM	1
PEFS_SMALL	2
Graph Plus Table	
PEGPT_GRAPH	0
PEGPT_TABLE	1
PEGPT_BOTH	2
Grid Line Control	
PEGLC_BOTH	0
PEGLC_YAXIS	1
PEGLC_XAXIS	2
PEGLC_NONE	3
Hot-Spot Type	
PEHS_NONE	0
PEHS_SUBSET	1
PEHS_POINT	2
PEHS_GRAPH	3
PEHS_TABLE	4
PEHS_DATAPOINT	5
Line Types	
PELT_THINSOLID	0
PELT_DASH	1
PELT_DOT	2
PELT_DASHDOT	3
PELT_DASHDOTDOT	4
PELT_MEDIUMSOLID	5
PELT_THICKSOLID	6
Manual Scale Control	
PEMSC_NONE	0
PEMSC_MIN	1
PEMSC_MAX	2
PEMSC_MINMAX	3
Object Types	
PECONTROL_GRAPH	300
PECONTROL_PIE	302

PECONTROL_SGRAPH 304

Plotting Methods - Graph Object

PEGPM_LINE 0
PEGPM_BAR 1
PEGPM_POINT 2
PEGPM_AREA 3
PEGPM_AREASTACKED 4
PEGPM_AREASTACKEDPERCENT 5
PEGPM_BARSTACKED 6
PEGPM_BARSTACKEDPERCENT 7
PEGPM_POINTPLUSBFL 8
PEGPM_POINTSPLUSBFLGRAPHED 9
PEGPM_HISTOGRAM 10

Plotting Methods - Scientific Graph Object

PESGPM_LINE 0
PESGPM_POINT 1
PESGPM_STICK 2
PESGPM_POINTPLUSBFL 3

Point Sizes

PEPS_SMALL 0
PEPS_MEDIUM 1
PEPS_LARGE 2

Point Types

PEPT_PLUS 0
PEPT_CROSS 1
PEPT_DOT 2
PEPT_DOTSOLID 3
PEPT_SQUARE 4
PEPT_SQUARESOLID 5
PEPT_DIAMOND 6
PEPT_DIAMONDSOLID 7
PEPT_UPTRIANGLE 8
PEPT_UPTRIANGLESOLID 9
PEPT_DOWNTRIANGLE 10
PEPT_DOWNTRIANGLESOLID 11

Points To Graph Initially

PEPTGI_FIRSTPOINTS 0
PEPTGI_LASTPOINTS 1

Points To Graph Version

PEPTGV_SEQUENTIAL 0
PEPTGV_RANDOM 1

Specific Plot Mode

PESPM_NONE 0
PESPM_HIGHLOWBAR 1
PESPM_HIGHLOWLINE 2
PESPM_HIGHLOWCLOSE 3
PESPM_OPENHIGHLOWCLOSE 4
PESPM_BOXPLOT 5

Table What

PETW_GRAPHED	0
PETW_ALLSUBSETS	1

Vertical Boundary

PEVB_NONE	0
PEVB_TOP	1
PEVB_BOTTOM	2
PEVB_TOPANDBOTTOM	3

Viewing Style

PEVS_COLOR	0
PEVS_MONO	1
PEVS_MONOWITHSYMBOLS	2

Customization Dialog Return IDs

IDEXPORTBUTTON	1015
IDMAXIMIZEBUTTON	1016
IDORIGINALBUTTON	1109

Messages

Notification Messages

<u>PEWN_CLICKED</u>	(WM_USER + 2930)
<u>PEWN_DBLCLICKED</u>	(WM_USER + 2931)
<u>PEWN_SETFOCUS</u>	(WM_USER + 2932)
<u>PEWN_KILLFOCUS</u>	(WM_USER + 2933)
<u>PEWN_CHANGINGPARMS</u>	(WM_USER + 2937)

Private Class Messages

<u>PEWM_PARENTVIEW_RESIZED</u>	(WM_USER + 2934)
--------------------------------	------------------

PEWN_CLICKED

The PEWN_CLICKED notification message is sent to the parent window when the user clicks the object. The object first gives itself the focus before sending this notification message.

Parameter	Description
wParam	Specifies the control identifier.
lParam	Contains a handle that identifies the object in its low-order word and the PEWN_CLICKED notification code in its high-order word.

Comments

This event can be used to receive a HOTSPOTDATA structure.

PEWN_DBLCLICKED

The PEWN_DBLCLICKED notification message is sent to the parent window when the user double-clicks the object. Before the notification message is sent, the object invokes the customization dialog or auto-coordinate prompting if PEP_bALLOWCUSTOMIZATION or PEP_bALLOWCOORDPROMPTING is TRUE.

Parameter	Description
wParam	Specifies the control identifier.
lParam	Contains a handle that identifies the object in its low-order word and the PEWN_DBLCLICKED notification code in its high-order word.

Comments

This event can be used to receive a HOTSPOTDATA structure.

PEWN_SETFOCUS

The PEWN_SETFOCUS notification message is sent to the parent window when the object receives WM_SETFOCUS. The object first invalidates the focus-rectangle before sending this notification message.

Parameter	Description
wParam	Specifies the control identifier.
lParam	Contains a handle that identifies the object in its low-order word and the PEWN_SETFOCUS notification code in its high-order word.

PEWN_KILLFOCUS

The PEWN_SETFOCUS notification message is sent to the parent window when the object receives WM_KILLFOCUS. The object first invalidates the focus-rectangle before sending this notification message.

Parameter	Description
wParam	Specifies the control identifier.
lParam	Contains a handle that identifies the object in its low-order word and the PEWN_KILLFOCUS notification code in its high-order word.

PEWN_CHANGINGPARMS

The PEWN_CHANGINGPARMS notification message is sent to the parent of the ProEssentials object when ever the PEreinitialize or PEreinitializcustoms functions are called.

The parent can use this message to know when object parameters have been adjusted.

Parameter	Description
wParam	Specifies the control identifier.
lParam	Contains a handle that identifies the object in its low-order word and the PEWN_CHANGINGPARMS notification code in its high-order word.

PEWM_PARENTVIEW_RESIZED

The PEWM_PARENTVIEW_RESIZED message is sent by a parent window to tell child ProEssentials objects to resize/reposition themselves at the logical location defined by PEP_rectLOGICALLOC with respect to the parents client logical extents which are represented by PEP_nPAGEHEIGHT and PEP_nPAGEWIDTH.

This message and the above properties are provided to help the developer implement auto-sizing functionality. Each child window must have the same values for PEP_nPAGEHEIGHT, and PEP_nPAGEWIDTH. The logical coordinate system size is left to the developer to decide.

Parameter	Description
wParam	Ignored
lParam	Ignored

Data Structures

GLOBALPROPERTIES
GRAPHPROPERTIES
SGRAPHPROPERTIES
PIEPROPERTIES

HOTSPOTDATA

GLOBALPROPERTIES

This structure holds properties common to all ProEssentials objects. The structure members are named and utilized just as the ProEssentials properties are named and utilized. This structure is used with the **PEsetglobal** and **PEgetglobal** functions.

```
typedef struct tagGLOBALPROPERTIES
{
    // GLOBAL SCOPE //
    UINT      nObjectType;
    char      szMainTitle[48];
    char      szSubTitle[48];
    UINT      nSubsets;
    UINT      nPoints;
    BOOL      bMonoWithSymbols;
    UINT      nDefOrientation;
    BOOL      bPrepareImages;
    BOOL      b3dDialogs;
    BOOL      bDataShadows;
    BOOL      bAllowCustomization;
    BOOL      bAllowExporting;
    BOOL      bAllowMaximization;
    BOOL      bAllowPopup;
    BOOL      bAllowUserInterface;
    UINT      nPageWidth;
    UINT      nPageHeight;
    RECT      rectLogicalLoc;
    BOOL      bCustom;
    BOOL      nViewingStyle;
    BOOL      nCViewingStyle;
    DWORD     dwMonoDeskColor;
    DWORD     dwMonoTextColor;
    DWORD     dwMonoShadowColor;
    DWORD     dwMonoGraphForeColor;
    DWORD     dwMonoGraphBackColor;
    DWORD     dwMonoTableForeColor;
    DWORD     dwMonoTableBackColor;
    DWORD     dwCMonoDeskColor;
    DWORD     dwCMonoTextColor;
    DWORD     dwCMonoShadowColor;
    DWORD     dwCMonoGraphForeColor;
    DWORD     dwCMonoGraphBackColor;
    DWORD     dwCMonoTableForeColor;
    DWORD     dwCMonoTableBackColor;
    DWORD     dwDeskColor;
    DWORD     dwTextColor;
    DWORD     dwShadowColor;
    DWORD     dwGraphForeColor;
    DWORD     dwGraphBackColor;
    DWORD     dwTableForeColor;
    DWORD     dwTableBackColor;
    DWORD     dwCDeskColor;
    DWORD     dwCTextColor;
    DWORD     dwCShadowColor;
    DWORD     dwCGraphForeColor;
```

```
DWORD    dwCGraphBackColor;
DWORD    dwCTableForeColor;
DWORD    dwCTableBackColor;
UINT     nDataPrecision;
UINT     nCDataPrecision;
UINT     nFontSize;
UINT     nCFontSize;
char     szMainTitleFont[48];
BOOL     bMainTitleBold;
BOOL     bMainTitleItalic;
BOOL     bMainTitleUnderline;
char     szCMainTitleFont[48];
BOOL     bCMainTitleBold;
BOOL     bCMainTitleItalic;
BOOL     bCMainTitleUnderline;
char     szSubTitleFont[48];
BOOL     bSubTitleBold;
BOOL     bSubTitleItalic;
BOOL     bSubTitleUnderline;
char     szCSubTitleFont[48];
BOOL     bCSubTitleBold;
BOOL     bCSubTitleItalic;
BOOL     bCSubTitleUnderline;
char     szLabelFont[48];
BOOL     bLabelBold;
BOOL     bLabelItalic;
BOOL     bLabelUnderline;
char     szCLabelFont[48];
BOOL     bCLabelBold;
BOOL     bCLabelItalic;
BOOL     bCLabelUnderline;
char     szTableFont[48];
char     szCTableFont[48];

// Version 1.5 //
BOOL     bAllowSubsetHotSpots;
BOOL     bAllowPointHotSpots;
} GLOBALPROPERTIES;
```

GRAPHPROPERTIES

This structure holds properties for the Graph Object. The structure members are named and utilized just as the ProEssentials properties are named and utilized. This structure is used with the **PEsetgraph** and **PEgetgraph** functions.

```
typedef struct tagGRAPHPROPERTIES
{
    // GRAPH SCOPE //
    char      szXAxisLabel[48];
    char      szYAxisLabel[48];
    UINT      nVBoundTypes;
    float     fUpperBoundValue;
    float     fLowerBoundValue;
    char      szUpperBoundText[48];
    char      szLowerBoundText[48];
    UINT      nYAxisScaleControl;
    UINT      nManualScaleControlY;
    float     fManualMinY;
    float     fManualMaxY;
    BOOL      bNoScrollingSubsetControl;
    UINT      nScrollingSubsets;
    UINT      nCScrollingSubsets;
    UINT      nPlottingMethod;
    UINT      nCPlottingMethod;
    UINT      nGridLineControl;
    UINT      nCGridLineControl;
    BOOL      bGridInFront;
    BOOL      bCGridInFront;
    BOOL      bTreatCompsAsNormal;
    BOOL      bCTreatCompsAsNormal;
    BOOL      bNoStackedData;
    UINT      nPointsToGraphInit;
    UINT      nPointsToGraphVersion;
    UINT      nCPointsToGraphVersion;
    UINT      nPointsToGraph;
    UINT      nCPointsToGraph;
    BOOL      bForceVerticalPoints;
    BOOL      bCForceVerticalPoints;
    UINT      nGraphPlusTable;
    UINT      nCGraphPlusTable;
    UINT      nTableWhat;
    UINT      nCTableWhat;

    // Version 1.5 //
    BOOL      bAllowDataHotSpots;
    BOOL      bAllowGraphHotSpots;
    BOOL      bAllowTableHotSpots;
    BOOL      bAllowCoordPrompting;
    BOOL      bMarkDataPoints;
    BOOL      bCMarkDataPoints;
    UINT      nRYAxisComparisonSubsets;
    UINT      nRYAxisScaleControl;
    UINT      nInitialScaleForRYData;
    UINT      nManualScaleControlRY;
}
```

```
float    fManualMinRY;
float    fManualMaxRY;
char     szRYAxisLabel[48];
BOOL     bAllowPlotCustomization;
BOOL     bAllowHistogram;
int      nSpecificPlotMode;
BOOL     bNegativeFromXAxis;
BOOL     bManualYAxisTicknLine;
float    fManualYAxisTick;
float    fManualYAxisLine;
BOOL     bManualRYAxisTicknLine;
float    fManualRYAxisTick;
float    fManualRYAxisLine;
int      nTargetPointsToTable;
int      nAltFreqThreshold;
float    fNullDataValue;
int      nPointSize;
BOOL     bAllowBestFitCurve;
int      nBestFitDegree;
BOOL     bAllowSpline;
int      nCurveGranularity;
float    fManualStackedMaxY;
int      nMaxPointsToGraph;
BOOL     bNoRandomPointsToGraph;
BOOL     bAllowLine;
BOOL     bAllowBar;
BOOL     bAllowPoint;
BOOL     bAllowBestFitLine;
BOOL     bAllowArea;
int      nAllowZooming;
BOOL     bForceRightYAxis;
BOOL     bAllowPointsPlusLine;
BOOL     bAllowPointsPlusSpline;
BOOL     bAllowBestFitLineII;
BOOL     bAllowBestFitCurveII;
int      nSymbolFrequency;
BOOL     bAppendToEnd;
} GRAPHPROPERTIES;
```

SGRAPHPROPERTIES

This structure holds properties for the Scientific Graph Object. The structure members are named and utilized just as the ProEssentials properties are named and utilized. This structure is used with the **PEsetsgraph** and **PEgetsgraph** functions.

```
typedef struct tagSGRAPHPROPERTIES
{
    // SCIENTIFIC GRAPH SCOPE //
    char      szXAxisLabel[48];
    char      szYAxisLabel[48];
    UINT      nVBoundTypes;
    float     fUpperBoundValue;
    float     fLowerBoundValue;
    char      szUpperBoundText[48];
    char      szLowerBoundText[48];
    UINT      nYAxisScaleControl;
    UINT      nManualScaleControlY;
    float     fManualMinY;
    float     fManualMaxY;
    BOOL      bNoScrollingSubsetControl;
    UINT      nScrollingSubsets;
    UINT      nCScrollingSubsets;
    UINT      nPlottingMethod;
    UINT      nCPlottingMethod;
    UINT      nGridLineControl;
    UINT      nCGridLineControl;
    BOOL      bGridInFront;
    BOOL      bCGridInFront;
    BOOL      bTreatCompsAsNormal;
    BOOL      bCTreatCompsAsNormal;
    UINT      nXAxisScaleControl;
    UINT      nManualScaleControlX;
    float     fManualMinX;
    float     fManualMaxX;
    BOOL      bGraphDataLabels;
    BOOL      bCGraphDataLabels;

    // Version 1.5 //
    BOOL      bAllowDataHotSpots;
    BOOL      bAllowGraphHotSpots;
    BOOL      bAllowCoordPrompting;
    BOOL      bMarkDataPoints;
    BOOL      bCMarkDataPoints;
    UINT      nRYAxisComparisonSubsets;
    UINT      nRYAxisScaleControl;
    UINT      nInitialScaleForRYData;
    UINT      nManualScaleControlRY;
    float     fManualMinRY;
    float     fManualMaxRY;
    char      szRYAxisLabel[48];
    BOOL      bAllowPlotCustomization;
    BOOL      bAllowBubble;
    int       nBubbleSize;
    BOOL      bNegativeFromXAxis;
}
```

```
int      nAllowDataLabels;
BOOL     bManualXAxisTicknLine;
float    fManualXAxisTick;
float    fManualXAxisLine;
BOOL     bManualYAxisTicknLine;
float    fManualYAxisTick;
float    fManualYAxisLine;
BOOL     bManualRYAxisTicknLine;
float    fManualRYAxisTick;
float    fManualRYAxisLine;
float    fNullDataValue;
int      nPointSize;
BOOL     bAllowBestFitCurve;
int      nBestFitDegree;
BOOL     bAllowSpline;
int      nCurveGranularity;
BOOL     bAllowLine;
BOOL     bAllowPoint;
BOOL     bAllowBestFitLine;
BOOL     bAllowStick;
int      nAllowZooming;
BOOL     bZoomMode;
float    fZoomMinX;
float    fZoomMaxX;
float    fZoomMinY;
float    fZoomMaxY;
BOOL     bForceRightYAxis;
BOOL     bAllowPointsPlusLine;
BOOL     bAllowPointsPlusSpline;
int      nSymbolFrequency;
} SGRAPHPROPERTIES;
```

PIEPROPERTIES

This structure holds properties for the Pie Chart Object. The structure members are named and utilized just as the ProEssentials properties are named and utilized. This structure is used with the **PEsetpie** and **PEgetpie** functions.

```
typedef struct tagPIEPROPERTIES
{
    UINT    nGroupingPercent;
    UINT    nCGroupingPercent;
    UINT    nDataLabelType;
    UINT    nCDataLabelType;
    UINT    nAutoExplode;
} PIEPROPERTIES;
```

HOTSPOTDATA

This structure holds parameters defining the most current hot-spot event. The ProEssentials reset this structure every time a WM_LBUTTONDOWN message is received.

```
typedef struct tagHOTSPOTDATA
{
    RECT    rectHotSpot;
    UINT    nHotSpotType;
    DWORD   dw1;
    DWORD   dw2;
} HOTSPOTDATA;
```

Parameter	Description
<i>rectHotSpot</i>	This parameter can be ignored. It is used internally.
<i>nHotSpotType</i>	This parameter defines the type of hot-spot.
PEHS_NONE	
PEHS_SUBSET	dw1 = subset index
PEHS_POINT	dw1 = point index**
PEHS_DATAPOINT	dw1 = subset index, dw2 = point index
PEHS_GRAPH	dw1 = x value, dw2 = y value
PEHS_TABLE	dw1 = subset index, dw2 = point index

The SDK developer will recognize Hot-Spots events by responding to the PEWN_CLICKED or PEWN_DBLCLICKED notification messages via WM_COMMAND. In response to a WM_COMMAND with a ProEssentials notification message, use PEvget to receive the current HOTSPOTDATA structure. If *nHotSpotType* does not equal PEHS_NONE, then a Hot-Spot event has occurred. The developer then uses *dw1* and *dw2* to further define what type of event has occurred.

** A PEHS_POINT event for the Pie Chart Object will also give the subset index in the dw2 structure member.

Also see PEP_structHOTSPOTDATA.

VBX Reference

[Global VBX Standard Properties](#)
[Global ProEssentials Properties](#)

[Graph Properties](#)
[Scientific Graph Properties](#)
[Pie Chart Properties](#)

[Global VBX Standard Events](#)
[Global ProEssentials Events](#)

[Global VBX Standard Methods](#)

Global VBX Standard Properties

The following are the standard Visual Basic properties that ProEssentials objects provide. Please refer to the Visual Basic documentation for information on these features.

Align, (2.0)	BorderStyle (off)
CtlName / Name	DragIcon
DragMode	Enabled
Height	HelpContextID, (2.0)
hWnd	Index
Left	MousePointer
Parent	TabIndex
TabStop	Tag
Top	Visible
Width	

Global ProEssentials VBX Custom Properties

The following are those custom properties belonging to all ProEssentials objects that are included into the VBX interfaces. Information about these properties can be found in the Property Reference section of this manual. The properties in the reference section all start with PEP_x where x represents the variable type of the property.

*** Note:** There are five properties that are an exception: **hObject**, **PEactions**, **PEnerg1**, **PEnerg2**, and **PEstrarg1**

hObject is a read only property that specifies the handle to the ProEssentials object being managed by the VBX interface. Visual Basic developers won't normally use this property. Visual C++ developers can use this property in ProEssentials DLL API calls to bypass the VBX interface. All VBX interfaces define this property as the default property.

PEactions, **PEnerg1**, **PEnerg2**, and **PEstrarg1** are properties which allow the developer to perform various actions.

For the following actions that call ProEssentials API Functions that need arguments, use:

PEnerg1 to set the width argument

PEnerg2 to set the height argument

PEstrarg1 to set the path\filename argument

* Be sure to initialize these argument properties before setting the PEactions property.

Code

PEactions = 0

PEactions = 1

PEactions = 2

PEactions = 3

PEactions = 4

PEactions = 5

PEactions = 6

PEactions = 7

PEactions = 8

PEactions = 9

PEactions = 10

PEactions = 11

PEactions = 12

PEactions = 13

PEactions = 14

PEactions = 15

PEactions = 16

PEactions = 17

Action

calls [PEreinitialize](#), [PEresetimage](#), and [InvalidateRect](#)

calls [PEreinitialize](#)

calls [PEresetimage](#)

calls the Windows [InvalidateRect](#) function to cause the control to redraw itself

calls [PElaunchmaximize](#)

calls [PElaunchcustomize](#)

calls [PElaunchexport](#)

calls [PElaunchtextexport](#), use PEnerg1 to represent bToFile 0=false, 1=true

calls [PElaunchprintdialog](#)

calls [PElaunchcolordialog](#)

calls [PElaunchfontdialog](#)

calls [PElaunchpopupmenu](#)

calls [PEcopymetatoclipboard](#)

calls [PEcopymetatofile](#)

calls [PEcopybitmaptoclipboard](#)

calls [PEcopybitmaptofile](#)

calls [PEcopyoletoclipboard](#)

calls [PEreinitializecustoms](#)

[3dDialogs](#)

[AllowCustomization](#)

[AllowExporting](#)

[AllowMaximization](#)

[AllowPointHotSpots](#)

[AllowPopup](#)

[AllowSubsetHotSpots](#)

[AllowUserInterface](#)

[DataPrecision](#)

[DataShadows](#)

[DefOrientation](#)

DeskColor
FontSize
GraphBackColor
GraphForeColor
hObject
LableBold
LabelFont
LabelItalic
LabelUnderline
MainTitle
MainTitleBold
MainTitleFont
MainTitleItalic
MainTitleUnderline
MonoWithSymbols
ObjectType
Points
PrepareImages
ShadowColor
SubsetLabels
Subsets
SubTitle
SubTitleBold
SubTitleFont
SubTitleItalic
SubTitleUnderline
TextColor
ViewingStyle
YData

Graph VBX Custom Properties

The following are those custom properties belonging to the Graph Object that are included into the VBX interface PEGRPH.VBX. Information about these properties can be found in the Property Reference section of this manual. The properties in the reference section all start with PEP_x where x represents the variable type of the property.

[AllowArea](#)
[AllowBar](#)
[AllowBestFitCurve](#)
[AllowBestFitCurvell](#)
[AllowBestFitLine](#)
[AllowBestFitLinell](#)
[AllowCoordPrompting](#)
[AllowDataHotSpots](#)
[AllowGraphHotSpots](#)
[AllowHistogram](#)
[AllowLine](#)
[AllowPlotCustomization](#)
[AllowPoint](#)
[AllowPointHotSpots](#)
[AllowPointsPlusLine](#)
[AllowPointsPlusSpline](#)
[AllowSpline](#)
[AllowTableHotSpots](#)
[AllowZooming](#)
[BestFitDegree](#)
[AltFreqThreshold](#)
[AltFrequencies](#)
[AppendToEnd](#)
[AutoStatSubsets](#)
[ComparisonSubsets](#)
[CurveGranularity](#)
[ForceRightYAxis](#)
[ForceVerticalPoints](#)
[GraphPlusTable](#)
[GridInFront](#)
[GridLineControl](#)
[LowerBoundText](#)
[LowerBoundValue](#)
[ManualMaxDataString](#)
[ManualMaxPointLabel](#)
[ManualMaxRY](#)
[ManualMaxY](#)
[ManualMinRY](#)
[ManualMinY](#)
[ManualRYAxisLine](#)
[ManualRYAxisTick](#)
[ManualRYAxisTicknLine](#)
[ManualScaleControlRY](#)
[ManualScaleControlY](#)
[ManualStackedMaxY](#)
[ManualYAxisLine](#)
[ManualYAxisTick](#)

ManualYAxisTicknLine
MaxPointsToGraph
MarkDataPoints
NegativeFromXAxis
NoRandomPointsToGraph
NoScrollingSubsetControl
NullDataValue
PlottingMethod
PointSize
PointLabels
PointsToGraph
RandomPointsToGraph
RandomSubsetsToGraph
RYAxisComparisonSubsets
RYAxisLabel
RYAxisScaleControl
ScaleForRYData
ScrollingSubsets
SpecificPlotMode
SubsetLineTypes
SubsetPointTypes
SymbolFrequency
TableBackColor
TableFont
TableForeColor
TableWhat
TargetPointsToTable
TreatCompsAsNormal
UpperBoundText
UpperBoundValue
VBoundaryTypes
XAxisLabel
YAxisLabel
YAxisScaleControl

Scientific Graph VBX Custom Properties

The following are those custom properties belonging to the Scientific Graph Object that are included into the VBX interface PESGRPH.VBX. Information about these properties can be found in the Property Reference section of this manual. The properties in the reference section all start with PEP_x where x represents the variable type of the property.

[AllowBestFitCurve](#)
[AllowBestFitLine](#)
[AllowBubble](#)
[AllowCoordPrompting](#)
[AllowDataHotSpots](#)
[AllowDataLabels](#)
[AllowGraphHotSpots](#)
[AllowLine](#)
[AllowPlotCustomization](#)
[AllowPoint](#)
[AllowPointsPlusLine](#)
[AllowPointsPlusSpline](#)
[AllowSpline](#)
[AllowStick](#)
[AllowZooming](#)
[BestFitDegree](#)
[BubbleSize](#)
[ComparisonSubsets](#)
[CurveGranularity](#)
[DataPointLabels](#)
[ForceRightYAxis](#)
[GraphDataLabels](#)
[GridInFront](#)
[GridLineControl](#)
[LowerBoundText](#)
[LowerBoundValue](#)
[ManualMaxRY](#)
[ManualMaxX](#)
[ManualMaxY](#)
[ManualMinRY](#)
[ManualMinX](#)
[ManualMinY](#)
[ManualRYAxisLine](#)
[ManualRYAxisTick](#)
[ManualRYAxisTicknLine](#)
[ManualScaleControlRY](#)
[ManualScaleControlX](#)
[ManualScaleControlY](#)
[ManualYAxisLine](#)
[ManualYAxisTick](#)
[ManualYAxisTicknLine](#)
[MarkDataPoints](#)
[NegativeFromXAxis](#)
[NoScrollingSubsetControl](#)
[NullDataValue](#)
[PlottingMethod](#)
[PointSize](#)

RandomSubsetsToGraph
RYAxisComparisonSubsets
RYAxisLabel
RYAxisScaleControl
ScaleForRYData
ScrollingSubsets
SubsetLineTypes
SubsetPointTypes
SymbolFrequency
TreatCompsAsNormal
UpperBoundText
UpperBoundValue
VBoundaryTypes
XAxisLabel
XAxisScaleControl
YAxisLabel
YAxisScaleControl
XData
ZData
ZoomMaxX
ZoomMaxY
ZoomMinX
ZoomMaxY
ZoomMode

Pie Chart VBX Custom Properties

The following are those custom properties belonging to the Pie Chart Object that are included into the VBX interface PEPIE.VBX. Information about these properties can be found in the Property Reference section of this manual. The properties in the reference section all start with PEP_x where x represents the variable type of the property.

AutoExplode

DataLabelType

GroupingPercent

PointLabels

XData

Global VBX Standard Events

The following are the standard Visual Basic events that ProEssentials objects provide. Please refer to the Visual Basic documentation for information on these features.

Click	DbIcIck
DragDrop	DragOver
GotFocus	KeyDown
KeyPress	KeyUp
LostFocus	MouseDown
MouseMove	MouseUp

Global ProEssentials VBX Custom Events

The following are those custom events belonging to ProEssentials objects.

VertScroll

ParamUpdate

SubsetHotSpot

PointHotSpot

DataHotSpot

GraphHotSpot

TableHotSpot

VertScroll VBX Event

Scope All ProEssentials VBXs

Sub `ctlname_VertScroll(SBCode%, SBPosition%)`

The VertScroll event is fired after the object is scrolled either by 1) the user clicking the vertical scroll bar or 2) the user pressing the up or down arrow keys.

Parameter	Description
<i>SBCode</i>	Windows defined scrollbar codes. SB_BOTTOM=7 SB_LINEDOWN=1 SB_LINEUP=0 SB_PAGEDOWN=3 SB_PAGEUP=2 SB_THUMBPOSITION=4 SB_THUMBTRACK=5 SB_TOP=6
<i>SBPosition</i>	Current position of scrollbar.

Comments

This event allows the ProEssentials object to function as a vertical scrollbar, and thus control other related subset information. If you are using the ProEssentials object as a scrollbar, you should set `NoScrollingSubsetControl = TRUE`, and `ScrollingSubsets = 1`.

ParamUpdate VBX Event

Scope All ProEssentials VBXs

Sub `ctlname_ParamUpdate()`

The ParamUpdate event is fired after the **PREinitialize**, or **PREinitializecustoms** functions are executed. This event can be used to recognize changes to scrollbar ranges and other properties.

SubsetHotSpot VBX Event

Scope All ProEssentials VBXs

Sub ctlname_**SubsetHotSpot**(*DbIClk%*, *SubsetIndex%*)

The SubsetHotSpot event is fired when the user clicks a subset label.

Parameter	Description
<i>DbIClk</i>	0 = Click 1 = Double Click

<i>SubsetIndex</i>	Zero based index of subset label clicked.
--------------------	---

Comments

This event is only fired if PEP_bALLOWSSUBSETHOTSPOTS is TRUE.

PointHotSpot VBX Event

Scope Graph

Sub `ctlname_PointHotSpot(DbIClk%, PointIndex%)`

Scope PieChart

Sub `ctlname_PointHotSpot(DbIClk%, SubsetIndex%, PointIndex%)`

The PointHotSpot event is fired when the user clicks a point label.

Parameter	Description
<i>DbIClk</i>	0 = Click 1 = Double Click
<i>PointIndex</i>	Zero based index of point label clicked.
<i>SubsetIndex</i>	Zero based index of currently shown subset. This allows the PointHotSpot event to function like a DataHotSpot event.

Comments

This event is only fired if PEP_bALLOWPOINTHOTSPOTS is TRUE.

DataHotSpot VBX Event

Scope Graph and Scientific Graph VBXs

Sub `ctlname_DataHotSpot(DbIClk%, SubsetIndex%, PointIndex%)`

The DataHotSpot event is fired when the user clicks a data point.

Parameter	Description
<i>DbIClk</i>	0 = Click 1 = Double Click
<i>SubsetIndex</i>	Zero based index of subset clicked.
<i>PointIndex</i>	Zero based index of point clicked.

Comments

This event is only fired if PEP_bALLOWDATAHOTSPOTS is TRUE.

GraphHotSpot VBX Event

Scope Graph and Scientific Graph VBXs

Sub `ctlname_GraphHotSpot(DbClk%, XVal!, YVal!)`

The GraphHotSpot event is fired when the user clicks inside the graphs grid.

Parameter	Description
<i>DbClk</i>	0 = Click 1 = Double Click
<i>XVal</i>	Grid X Axis Coordinate.
<i>YVal</i>	Grid Y Axis Coordinate.

Comments

This event is only fired if PEP_bALLOWGRAPHHOTSPOTS is TRUE.

TableHotSpot VBX Event

Scope Graph VBX

Sub `ctlname_TableHotSpot(DbIClk%, SubsetIndex%, PointIndex%)`

The TableHotSpot event is fired when the user clicks inside the table of a Graph Object.

Parameter	Description
<i>DbIClk</i>	0 = Click 1 = Double Click
<i>SubsetIndex</i>	Zero based index of subset clicked.
<i>PointIndex</i>	Zero based index of point clicked.

Comments

This event is only fired if PEP_bALLOWTABLEHOTSPOTS is TRUE.

Global ProEssentials VBX Methods

The ProEssentials VBX interfaces offers the following methods:

Drag
Move
Refresh
SetFocus
ZOrder

All these methods function in the Visual Basic default behavior except Refresh. The Refresh method calls the **PEreinitialize** and **PEresetimage** functions and then invalidates and updates the objects image.

