PascalScript - syntax

Here is the brief overview of PascalScript language syntax, that is implemented in the Precision Builder Express product. The help for the reserved words, types, objects, functions and other identifiers of Pascal language can be found in the Delphi or in the FreePascal help references, or on the web. The syntax of objects, methods and identifiers, that are specific to the Precision Builder Express, is described in this topic.

For the PascalScript language implementation the third-party components have been used (RemObjects Software PascalScript) , so when writting the code, it is necessary to take into account some specifics in compare to code written with Delphi or FreePascal tools. Detailed description of scripting language possibilities can be found on the manufacturer's web site: http://www.remobjects.com/ps.aspx, but it is a good practice to use a "**code completion**" feature of the Code Editor, that shows the list of available objects, methods, types, variables and constants.

Reserved words

and	**except**	**of**	**uses**
array	**exports**	**on**	**var**
as	**finally**	**or**	**while**
begin	**for**	**packed**	**with**
case	**function**	**procedure**	**xor**
const	**if**	**raise**	**string**
div	**in**	**record**	**then**
do	**is**	**repeat**	**to**
downto	**mod**	**set**	**try**
else	**nil**	**shl**	**type**
end	**not**	**shr**	**until**

Types

Simple types

- AnsiChar
- AnsiString
- AnyString = String
- Boolean
- Byte
- ByteBool
- Cardinal
- Currency
- Double
- Extended
- Int64
- Integer
- LongBool
- LongInt
- LongWord
- NativeString
- PAnsiChar
- PChar
- ShortInt
- Single
- SmallInt
- String
- tbtString = String
- Variant
- Char
- Word
- WordBool

Windows data types

- HACCEL = Cardinal
- HBITMAP = Integer
- HMenu = Cardinal
- HPALETTE = Integer
- HResult = LongInt
- HWND = LongInt
- IDispatch = IUnknown
- TCLSID = TGUID
- TGUID = record D1: LongWord; D2: Word; D3: Word; D4: array[0..7] of Byte; end;
- THandle = LongInt
- TIID = TGUID

Other types

- EMenuError = Exception
- TAlign = (alNone, alTop, alBottom, alLeft, alRight, alClient)
- TAlignment = (taLeftJustify, taRightJustify, taCenter)

- TAnchorKind = (akTop, akLeft, akRight, akBottom)
- TAnchors = set of TAnchorKind
- TBevelShape = (bsBox, bsFrame, bsTopLine, bsBottomLine, bsLeftLine, bsRightLine,bsSpacer)
- TBevelStyle = (bsLowered, bsRaised)
- TBevelWidth = LongInt
- TBorderIcon = (biSystemMenu, biMinimize, biMaximize, biHelp)
- TBorderIcons = set of TBorderIcon
- TBorderStyle = TFormBorderStyle
- TBorderWidth = LongInt
- TBrushStyle = (bsSolid, bsClear, bsHorizontal, bsVertical, bsFDiagonal, bsBDiagonal, bsCross, bsDiagCross)
- TCloseAction = (caNone, caHide, caFree, caMinimize)
- TCMenuItem = TMenuItem
- TColor = integer
- TComboBoxStyle = (csDropDown, csSimple, csDropDownList, csOwnerDrawFixed, csOwnerDrawVariable)
- TComponentState = set of TComponentStateE
- TComponentStateE = (csLoading, csReading, csWriting, csDestroying, csDesigning, csAncestor, csUpdating, csFixups, csFreeNotification, csInline, csDesignInstance)
- TCursor = Integer
- TCustomDrawStage = (cdPrePaint, cdPostPaint, cdPreErase, cdPostErase)
- TCustomDrawState = set of TCustomDrawStateEl
- TCustomDrawStateEl = (cdsSelected, cdsGrayed, cdsDisabled, cdsChecked, cdsFocused, cdsDefault, cdsHot, cdsMarked, cdsIndeterminate)
- TCustomDrawTarget = (dtControl, dtItem, dtSubItem)
- TDateTime = Extended
- TDisplayCode = (drBounds, drIcon, drLabel, drSelectBounds)
- TDragKind = (dkDrag, dkDock)
- TDragMode = (dmManual, dmAutomatic)
- TDragState = (dsDragEnter, dsDragLeave, dsDragMove)
- TDuplicates = (dupIgnore, dupAccept, dupError)
- TEditCharCase = (ecNormal, ecUpperCase, ecLowerCase)
- TEOwnerDrawState = (odSelected, odGrayed, odDisabled, odChecked, odFocused, odDefault, odHotLight, odInactive, odNoAccel, odNoFocusRect, odReserved1, odReserved2, odComboBoxEdit)
- TEShiftState = (ssShift, ssAlt, ssCtrl, ssLeft, ssRight, ssMiddle, ssDouble)
- TFindItemKind = (fkCommand, fkHandle, fkShortcut)
- TFontPitch = (fpDefault, fpVariable, fpFixed)
- TFontStyle = (fsBold, fsItalic, fsUnderline, fsStrikeOut)
- TFontStyles = set of TFontStyle
- TFormBorderStyle = (bsNone, bsSingle, bsSizeable, bsDialog, bsToolWindow, bsSizeToolWin)
- TFormStyle = (fsNormal, fsMDIChild, fsMDIForm, fsStayOnTop)
- THelpContext = LongInt
- THitTest = (htAbove, htBelow, htNowhere, htOnItem, htOnButton, htOnIcon, htOnIndent, htOnLabel, htOnRight, htOnStateIcon, htToLeft, htToRight)
- THitTests = set of THitTest
- TCheckBoxState = (cbUnchecked, cbChecked, cbGrayed)
- TIconArrangement = (iaTop, iaLeft)
- TIFException = (ErNoError, erCannotImport, erInvalidType, ErInternalError, erInvalidHeader, erInvalidOpcode, erInvalidOpcodeParameter, erNoMainProc, erOutOfGlobalVarsRange, erOutOfProcRange, ErOutOfRange, erOutOfStackRange, ErTypeMismatch, erUnexpectedEof, erVersionError, ErDivideByZero, ErMathError,erCouldNotCallProc, erOutofRecordRange, erOutOfMemory, erException, erNullPointerException, erNullVariantError, erInterfaceNotSupported, erCustomError)
- TItemFind = (ifData, ifPartialString, ifExactString, ifNearest)
- TItemChange = (ctText, ctImage, ctState)
- TItemRequest = set of TItemRequests
- TItemRequests = (irText, irImage, irParam, irState, irIndent)
- TItemState = (isNone, isCut, isDropHilited, isFocused, isSelected, isActivating)
- TItemStates = set of TItemState
- TListArrangement = (arAlignBottom, arAlignLeft, arAlignRight, arAlignTop, arDefault, arSnapToGrid)
- TListBoxStyle = (lbStandard, lbOwnerDrawFixed, lbOwnerDrawVariable)
- TListHotTrackStyle = (htHandPoint, htUnderlineCold, htUnderlineHot)
- TListHotTrackStyles = set of TListHotTrackStyle
- TLVAdvancedCustomDrawEvent = procedure (Sender : TCustomListView; const ARect : TRect; Stage : TCustomDrawStage; var DefaultDraw : Boolean)
- TLVAdvancedCustomDrawItemEvent = procedure (Sender : TCustomListView; Item : TListItem; State : TCustomDrawState; Stage : TCustomDrawStage; var DefaultDraw : Boolean)
- TLVAdvancedCustomDrawSubItemEvent = procedure (Sender : TCustomListView; Item : TListItem; SubItem : Integer; State : TCustomDrawState; Stage : TCustomDrawStage; var DefaultDraw : Boolean)
- TLVColumnClickEvent = procedure (Sender : TObject; Column : TListColumn)
- TLVColumnRClickEvent = procedure (Sender : TObject; Column : TListColumn; Point : TPoint)
- TLVCompareEvent = procedure (Sender : TObject; Item1, Item2 : TListItem; Data : Integer; var Compare : Integer)
- TLVCustomDrawEvent = procedure (Sender : TCustomListView; const ARect : TRect; var DefaultDraw : Boolean)
- TLVCustomDrawItemEvent = procedure (Sender : TCustomListView; Item : TListItem; State : TCustomDrawState; var DefaultDraw : Boolean)
- TLVCustomDrawSubItemEvent = procedure (Sender : TCustomListView; Item : TListItem; SubItem : Integer; State : TCustomDrawState; var DefaultDraw : Boolean)
- TLVDeletedEvent = procedure (Sender : TObject; Item : TListItem)

- TLVDrawItemEvent = procedure (Sender : TCustomListView; Item : TListItem; Rect : TRect; State : TOwnerDrawState)
- TLVEditedEvent = procedure (Sender : TObject; Item : TListItem; var S : string)
- TLVEditingEvent = procedure (Sender : TObject; Item : TListItem; var AllowEdit : Boolean)
- TLVChangeEvent = procedure (Sender : TObject; Item : TListItem; Change : TItemChange)
- TLVChangingEvent = procedure (Sender : TObject; Item : TListItem; Change : TItemChange; var AllowChange : Boolean)
- TLVInfoTipEvent = procedure (Sender : TObject; Item : TListItem; var InfoTip : string)
- TLVNotifyEvent = procedure (Sender : TObject; Item : TListItem)
- TLVOwnerDataEvent = procedure (Sender : TObject; Item : TListItem)
- TLVOwnerDataHintEvent = procedure (Sender : TObject; StartIndex, EndIndex : Integer)
- TLVOwnerDataStateChangeEvent = procedure (Sender : TObject; StartIndex, EndIndex : Integer; OldState, NewState : TItemStates)
- TLVSelectItemEvent = procedure (Sender : TObject; Item : TListItem; Selected : Boolean)
- TLVSubItemImageEvent = procedure (Sender : TObject; Item : TListItem; SubItem : Integer; var ImageIndex : Integer)
- TMeasureItemEvent = procedure(Control: TWinControl; Index: Integer; var Height: Integer)
- TMenuAnimation = set Of TMenuAnimations
- TMenuAnimations = (maLefttoright, maRightToLeft, maTopToBottom, maBottomToTop, maNone)
- TMenuAutoflag = TMenuItemAutoflag
- TMenuBreak = (mbNone, mbBreak, mbBarBreak)
- TMenuDrawitemevent = procedure (Sender : TObject; ACanvas : TCanvas; ARect : TRect; Selected : Boolean)
- TMenuChangeevent = procedure (Sender : TObject; Source : TMenuitem; Rebuild : Boolean)
- TMenuItemautoflag = (maAutomatic, maManual, maParent)
- TMenuMeasureitemevent = procedure (Sender : TObject; ACanvas : TCanvas; Var Width, Height : Integer)
- TModalResult = Integer
- TMouseButton = (mbLeft, mbRight, mbMiddle)
- TMouseEvent = procedure (Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
- TMouseMoveEvent = procedure(Sender: TObject; Shift: TShiftState; X, Y: Integer);
- TMsgDlgBtn = (mbYes, mbNo, mbOK, mbCancel, mbAbort, mbRetry, mbIgnore, mbAll, mbNoToAll, mbYesToAll, mbHelp, mbClose)
- TMsgDlgButtons = set of TMsgDlgBtn
- TMsgDlgType = (mtWarning, mtError, mtInformation, mtConfirmation, mtCustom)
- TOperation = (opInsert, opRemove)
- TOwnerDrawState = set of TEOwnerDrawState
- TPanelBevel = (bvNone, bvLowered, bvRaised,bvSpace)
- TPenMode = (pmBlack, pmWhite, pmNop, pmNot, pmCopy, pmNotCopy, pmMergePenNot, pmMaskPenNot, pmMergeNotPen, pmMaskNotPen, pmMerge, pmNotMerge, pmMask, pmNotMask, pmXor, pmNotXor)
- TPenStyle = (psSolid, psDash, psDot, psDashDot, psDashDotDot, psClear, psInsideFrame)
- TPoint = record x,y: LongInt; end;
- TPopupAlignment = (paLeft, paRight, paCenter)
- TPosition = (poDesigned, poDefault, poDefaultPosOnly, poDefaultSizeOnly, poScreenCenter, poDesktopCenter, poMainFormCenter, poOwnerFormCenter)
- TPrintScale = (poNone, poProportional, poPrintToFit)
- TRect = record Left, Top, Right, Bottom: Integer; end;
- TScrollBarInc = SmallInt
- TScrollBarKind = (sbHorizontal, sbVertical)
- TScrollCode = (scLineUp, scLineDown, scPageUp, scPageDown, scPosition, scTrack, scTop, scBottom, scEndScroll)
- TScrollEvent = procedure(Sender: TObject; ScrollCode: TScrollCode;var ScrollPos: Integer)
- TScrollStyle = (ssNone, ssHorizontal, ssVertical, ssBoth)
- TSearchDirection = (sdLeft, sdRight, sdAbove, sdBelow, sdAll)
- TSectionEvent = procedure(Sender: TObject; ASection, AWidth: Integer)
- TShapeType = (stRectangle, stSquare, stRoundRect, stRoundSquare, stEllipse, stCircle)
- TShiftState = set of TEShiftState
- TShortCut = WORD
- TStartDragEvent = procedure (Sender: TObject; var DragObject: TDragObject)
- TTextLayout = (tlTop, tlCenter, tlBottom)
- TTrackButton = (tbRightButton, tbLeftButton)
- TVariantArray = array of Variant;
- TVarType = Word
- TViewStyle = (vsIcon, vsSmallIcon, vsList, vsReport)
- TWidth = Integer
- TWindowState = (wsNormal, wsMinimized, wsMaximized)

Event types

- TAdvancedMenuDrawItemEvent = procedure (Sender : TObject; ACanvas : TCanvas; ARect : TRect; State : TOwnerDrawState);
- TCloseEvent = procedure(Sender: TObject; var Action: TCloseAction);
- TCloseQueryEvent = procedure(Sender: TObject; var CanClose: Boolean);
- TDragDropEvent = procedure(Sender, Source: TObject;X, Y: Integer);
- TDragOverEvent = procedure(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean);
- TDrawItemEvent = procedure(Control: TWinControl; Index: Integer; Rect: TRect; State: Byte);
- TEndDragEvent = procedure(Sender, Target: TObject; X, Y: Integer);
- TGetStrProc = procedure(const S: string);
- THelpEvent = function (Command: Word; Data: LongInt; var CallHelp: Boolean): Boolean;
- TIdleEvent = procedure (Sender: TObject; var Done: Boolean);

- TKeyEvent = procedure (Sender: TObject; var Key: Word; Shift: TShiftState);
- TKeyPressEvent = procedure(Sender: TObject; var Key: Char);
- TNotifyEvent = procedure (Sender: TObject);

Constants

Action result constants (see also the ScriptResult variable)

- **SCRIPT_OK** : LongInt = 0
- **SCRIPT_ERROR** : LongInt = 1
- **SCRIPT_WARNING** : LongInt = 2
- **SCRIPT_HINT** : LongInt = 3

Color constants

- cl3DDkShadow : Integer = cl3DDkShadow
- cl3DLight : Integer = cl3DLight
- clActiveBorder : Integer = clActiveBorder
- clActiveCaption : Integer = clActiveCaption
- clAppWorkSpace : Integer = clAppWorkSpace
- clAqua : Integer = $FFFF00
- clBackground : Integer = clBackground
- clBlack : Integer = $000000
- clBlue : Integer = $FF0000
- clBtnFace : Integer = clBtnFace
- clBtnHighlight : Integer = clBtnHighlight
- clBtnShadow : Integer = clBtnShadow
- clBtnText : Integer = clBtnText
- clCaptionText : Integer = clCaptionText
- clDefault : Integer = $20000000
- clDkGray : Integer = $808080
- clFuchsia : Integer = $FF00FF
- clGray : Integer = $808080
- clGrayText : Integer = clGrayText
- clGreen : Integer = $008000
- clHighlight : Integer = clHighlight
- clHighlightText : Integer = clHighlightText
- clInactiveBorder : Integer = clInactiveCaption
- clInactiveCaption : Integer = clInactiveCaption
- clInactiveCaptionText : Integer = clInactiveCaptionText
- clInfoBk : Integer = clInfoBk
- clInfoText : Integer = clInfoText
- clLime : Integer = $00FF00
- clLtGray : Integer = $C0C0C0
- clMaroon : Integer = $000080
- clMenu : Integer = clMenu
- clMenuText : Integer = clMenuText
- clNavy : Integer = $800000
- clNone : Integer = $1FFFFFFF
- clOlive : Integer = $008080
- clPurple : Integer = $800080
- clRed : Integer = $0000FF
- clScrollBar : Integer = clScrollBar
- clSilver : Integer = $C0C0C0
- clTeal : Integer = $808000
- clWhite : Integer = $FFFFFF
- clWindow : Integer = clWindow
- clWindowFrame : Integer = clWindowFrame
- clWindowText : Integer = clWindowText
- clYellow : Integer = $00FFFF

Cursors constants

- crAppStart : Integer = -19
- crArrow : Integer = -2
- crCross : Integer = -3
- crDefault : Integer = 0
- crDrag : Integer = -12
- crHandPoint : Integer = -21
- crHelp : Integer = -20
- crHourGlass : Integer = -11
- crHSplit : Integer = -14
- crIBeam : Integer = -4
- crMultiDrag : Integer = -16

- crNo : Integer = -18
- crNoDrop : Integer = -13
- crNone : Integer = -1
- crSizeAll : Integer = -22
- crSizeNESW : Integer = -6
- crSizeNS : Integer = -7
- crSizeNWSE : Integer = -8
- crSizeWE : Integer = -9
- crSQLWait : Integer = -17
- crUpArrow : Integer = -10
- crVSplit : Integer = -15

  Date and time constants
- DateDelta : Longint = 693594
- MSecPerDay : Longint = 86400000
- SecsPerDay : Longint = 86400

  File opening mode constants
- fmCreate : Longint = $FFFF
- fmOpenRead : Longint = 0
- fmOpenReadWrite : Longint = 2
- fmOpenWrite : Longint = 1
- fmShareCompat : Longint = 0
- fmShareDenyNone : Longint = $40
- fmShareDenyRead : Longint = $30
- fmShareDenyWrite : Longint = $20
- fmShareExclusive : Longint = $10

  Message dialog constants
- mbAbortIgnore : LongInt = mbAbort or mbIgnore
- mbAbortRetryIgnore : LongInt = mbAbort or mbRetry or mbIgnore
- mbOKCancel : LongInt = mbOK or mbCancel
- mbYesAllNoAllCancel : LongInt = mbYes or mbYesToAll or mbNo or mbNoToAll or mbCancel
- mbYesNo : LongInt = mbYes or mbNo
- mbYesNoCancel : LongInt = mbYes or mbNo or mbCancel

  Modal result constants
- mrAbort : Integer = 3
- mrAll : Integer = 8
- mrCancel : Integer = 2
- mrIgnore : Integer = 5
- mrNo : Integer = 7
- mrNone : Integer = 0
- mrNoToAll : Integer = 9
- mrOk : Integer = 1
- mrRetry : Integer = 4
- mrYes : Integer = 6
- mrYesToAll : Integer = 10

  Stream seek offset constants
- soFromBeginning : Longint = 0
- soFromCurrent : Longint = 1
- soFromEnd : Longint = 2

  TParser token constants
- toEOF : Char = #0
- toFloat : Char = #4
- toInteger : Char = #3
- toString : Char = #2
- toSymbol : Char = #1

  Variant type constants
- varAny : Word
- varArray : Word
- varBoolean : Word
- varByRef : Word
- varByte : Word
- varCurrency : Word
- varDate : Word
- varDispatch : Word
- varDouble : Word
- varEmpty : Word
- varError : Word
- varInt64 : Word
- varInteger : Word

- varLongWord : Word
- varNull : Word
- varOleStr : Word
- varShortInt : Word
- varSingle : Word
- varSmallInt : Word
- varStrArg : Word
- varString : Word
- varTypeMask : Word
- varUnknown : Word
- varUString : Word
- varVariant : Word
- varWord : Word

VCL classes

- TApplication : TComponent
- TBevel : TGraphicControl
- TBitmap : TGraphic
- TBits : TObject
- TBrush : TGraphicsObject
- TButton : TButtonControl
- TButtonControl : TWinControl
- TCanvas : TPersistent
- TCollection : TPersistent
- TCollectionItem : TPersistent
- TComboBox : TCustomComboBox
- TCommonCalendar : TWinControl
- TComponent : TPersistent
- TControl : TComponent
- TControlScrollBar : TPersistent
- TCustomComboBox : TWinControl
- TCustomControl : TWinControl
- TCustomEdit : TWinControl
- TCustomGroupBox : TCustomControl
- TCustomCheckox : TButtonControl
- TCustomIniFile : TObject
- TCustomLabel : TGraphicControl
- TCustomListBox : TWinControl
- TCustomListView : TObject
- TCustomListView : TWinControl
- TCustomMemo : TCustomEdit
- TCustomMemoryStream : TStream
- TCustomPanel : TCustomControl
- TCustomRadioGroup : TCustomGroupBox
- TCustomTabControl : TWinControl
- TDateTimePicker : TCommonCalendar
- TDragObject : TObject
- TEdit : TCustomEdit
- TFileStream : THandleStream
- TFont : TGraphicsObject
- TForm : TScrollingWinControl
- TGraphic : TPersistent
- TGraphicControl : TControl
- TGraphicsObject : TPersistent
- TGroupBox : TCustomGroupBox
- THandleStream : TStream
- THeader : TCustomControl
- TCheckBox : TCustomCheckBox
- TIconOptions : TPersistent
- TImage : TGraphicControl
- TIniFile : TCustomIniFile
- TLabel : TCustomLabel
- TListBox : TCustomListBox
- TListColumn : TCollectionItem
- TListColumns : TCollection
- TListColumns : TObject
- TListItem : TPersistent
- TListItems : TObject

- TListItems : TPersistent
- TListView : TCustomListView
- TMainMenu : TMenu
- TMemIniFile : TCustomIniFile
- TMemo : TCustomMemo
- TMemoryStream : TCustomMemoryStream
- TMenu : TComponent
- TMenuActionLink : TActionLink
- TMenuItem : TComponent
- TMonthCalendar : TCommonCalendar
- TObject
- TOwnedCollection : TCollection
- TPageControl : TCustomTabControl
- TPaintBox : TGraphicControl
- TPanel : TCustomPanel
- TParser : TObject
- TPen : TGraphicsObject
- TPersistent : TObject
- TPopupMenu : TMenu
- TRadioButton : TButtonControl
- TRadioGroup : TCustomRadioGroup
- TResourceStream : TCustomMemoryStream
- TScrollBar : TWinControl
- TScrollBox : TScrollingWinControl
- TScrollingWinControl : TWinControl
- TShape : TGraphicControl
- TStream : TObject
- TStringList : TStrings
- TStrings : TPersistent
- TTabSheet : TWinControl
- TTimer : TComponent
- TWinControl : TControl
- TWorkArea : TCollectionItem
- TWorkAreas : TOwnedCollection

Standard procedures and functions

This is a list of standard PascalScript procedures and functions, that are well known from Delphi or FreePascal environment. The list also contains some special methods of RemObjects PascalScript implementation. Precision Builder Express specific procedures and functions are listed in separate section below.

- function Abs(e : Extended) : Extended;
- function AnsiLowercase(s : String) : String;
- function AnsiToOem(const S: String): String;
- function AnsiUppercase(s : String) : String;
- function Assigned(var P):Boolean;
- function ClearDir(const Dir: string): Boolean;
- function Copy(s: AnyString; iFrom, iCount: Longint): AnyString;
- function Cos(e : Extended) : Extended;
- function CreateComObject(const ClassID: TGUID): IUnknown;
- function CreateOleObject(const ClassName: String): IDispatch;
- function Date: TDateTime;
- function DateTimeToUnix(D: TDateTime): Int64;
- function DateToStr(D: TDateTime): String;
- function DayOfWeek(const DateTime: TDateTime): Word;
- procedure Dec(var X:Ordinal);
- procedure DecodeDate(const DateTime: TDateTime; var Year, Month, Day: Word);
- procedure DecodeTime(const DateTime: TDateTime; var Hour, Min, Sec, MSec: Word);
- procedure Delete(var s: AnyString; ifrom, icount: LongInt);
- function DeleteDir(const Dir: string): Boolean;
- function DeleteFile(const FileName: string): Boolean;
- function DirectoryExists(const Directory: string): Boolean;
- function DLLGetLastError: Longint
- procedure DrawMenuItem(MenuItem : TMenuItem; ACanvas : TCanvas; ARect : TRect; State : TOwnerDrawState);
- function EncodeDate(Year, Month, Day: Word): TDateTime;
- function EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime;
- function ExceptionParam: String;
- function ExceptionPos: Cardinal;
- function ExceptionProc: Cardinal;
- function ExceptionToString(er: TIFException; Param: String): String;

- function ExceptionType: TIFException;
- function ExcludeTrailingPathDelimiter(const S: string): string;
- function ExpandFileName(const FileName: string): string;
- function ExtractFileName(const FileName: string): string;
- function ExtractFilePath(const FileName: string): string;
- function FileExists(const FileName: string): Boolean;
- function FindGlobalComponent(const Name: string): TComponent;
- function FindGlobalObject(Path:string):TObject;
- function FloatToStr(e : Extended) : String;
- function ForceDirectories(Dir: string): Boolean;
- function FormatDateTime(const fmt: String; D: TDateTime): String;
- function FormatFloat(const Format: string; Value: Extended): string;
- function GetActiveOleObject(const ClassName: String): IDispatch;
- function GetArrayLength(var A): Integer;
- function GetCurrentDir: string;
- function GetFileSize(const FileName: string): Longint;
- function High(var A): Int64;
- function High(var A): Integer;
- function ChangeFileExt(const FileName, Extension: string): string;
- function IDispatchInvoke(Self: IDispatch; PropertySet: Boolean; const Name: String; Par: array of variant): variant;
- procedure Inc(var X:Ordinal);
- function IncludeTrailingPathDelimiter(const S: string): string;
- function IncMonth(const Date: TDateTime; NumberOfMonths: Integer): TDateTime;
- function IncTimes(ATime: TDateTime; Hours, Minutes, Seconds, MSecs: Integer): TDateTime;
- function InputBox(const ACaption, APrompt, ADefault: string): string;
- function InputQuery(const ACaption, APrompt: string; var Value: string): Boolean;
- procedure Insert(s: AnyString; var s2: AnyString; iPos: LongInt);
- function Int(e : Extended) : Extended;
- function Int64ToStr(i: Int64): String;
- function IntToStr(i: Int64): String;
- function IntTostr(i: Integer): String;
- function IsLeapYear(Year: Word): Boolean;
- function Length(var A): Integer;
- function Low(var A): Int64;
- function Low(var A): Integer;
- function Lowercase(s : AnyString) : AnyString;
- function MessageDlg(const Msg : string; DlgType : TMsgDlgType; Buttons : TMsgDlgButtons; HelpCtx : Longint) : Integer
- function NewItem(const ACaption : String; AShortCut : TSshortCut; AChecked, AEnabled : Boolean; AOnClick : TNotifyEvent; hctx : Word; const AName : String) : TMenuItem;
- function NewLine : TMenuItem;
- function NewMenu(Owner : TComponent; const AName : String; Items : array of TMenuItem) : TMainMenu;
- function NewPopupMenu(Owner : TComponent; const AName : String; Alignment : TPopupAlignment; AutoPopup : Boolean; const Items : array of TCMenuItem) : TPopupMenu;
- function NewSubMenu(const ACaption : String; hctx : WORD; const AName : String; Items : array of TMenuItem; AEnabled : Boolean) : TMenuItem;
- function Now: TDateTime;
- function Null: Variant;
- function OemToAnsi(const S: String): String;
- procedure OleCheck(Result: HResult);
- function Padl(s : AnyString;l : LongInt) : AnyString;
- function Padr(s : AnyString;l : LongInt) : AnyString;
- function Padz(s : AnyString;l : LongInt) : AnyString;
- function Pi : Extended;
- function Pos(SubStr, S: AnyString): LongInt;
- function PromptForFileName(var AFileName: string; const AFilter: string; const ADefaultExt: string; const ATitle: string; const AInitialDir: string; SaveDialog: Boolean): Boolean;
- procedure RaiseException(Ex: TIFException; Param: String);
- procedure RaiseLastException;
- function Replicate(c : char;l : LongInt) : String;
- function Round(e : Extended) : LongInt;
- procedure SetArrayLength(var A; NewLength: Integer);
- function SetCurrentDir(const Dir: string): Boolean;
- procedure SetLength(var S; NewLength:Integer);
- function ShortCut(Key : Word; Shift : TShiftState) : TShortCut
- procedure ShortCutToKey(ShortCut : TShortCut; var Key : Word; var Shift : TShiftState);
- function ShortCutToText(Shortcut : TShortCut) : String;
- procedure ShowMessage(const Msg: string);
- function Sin(e : Extended) : Extended;
- function SizeOf(var X): LongInt;

- function Sqrt(e : Extended) : Extended;
- function StrGet(var S : String; I : Integer) : Char;
- function StrGet2(S : String; I : Integer) : Char;
- function StringOfChar(c : char;I : LongInt) : String;
- function StringToGUID(const S: string): TGUID;
- procedure StrSet(c : Char; I : Integer; var s : String);
- function StrToDate(const s: String): TDateTime;
- function StrToFloat(s: String): Extended;
- function StrToInt(s: String): LongInt;
- function StrToInt64(s: String): int64;
- function StrToIntDef(s: String; def: LongInt): LongInt;
- function TextToShortCut(Text : String) : TShortCut
- function Time: TDateTime;
- function Trim(s : AnyString) : AnyString;
- function Trunc(e : Extended) : LongInt;
- function TryEncodeDate(Year, Month, Day: Word; var Date: TDateTime): Boolean;
- function TryEncodeTime(Hour, Min, Sec, MSec: Word; var Time: TDateTime): Boolean;
- function Unassigned: Variant;
- function UnixToDateTime(U: Int64): TDateTime;
- procedure UnloadDll(s: string);
- function Uppercase(s : AnyString) : AnyString;
- function VarArrayGet(var S : Variant; I : Integer) : Variant;
- procedure VarArraySet(c : Variant; I : Integer; var s : Variant);
- function VarIsEmpty(const V: Variant): Boolean;
- function VarIsNull(const V: Variant): Boolean;
- function VarType(const V: Variant): TVarType;

Special components and variables

- **Application** : TApplication
- **MainForm** : TForm
  A direct reference to the main application form
- **ProjectVars** : TStringList
  This object represents a user defined variables of the current project. You can read, write, add or remove the variable values by using the properties and methods of this TStringList object (as for example: ProjectVars.Values['TEST']:='123';).
- **ScriptResult** : Integer
  This variable represents a Precision Builder Express action result. See the action result constants for available values.
- **ScriptVars** : TStringList
  This object represents a values from the definition graphic user interface of an action. For example, if you had defined a *TEdit* component *edFileName* in the definition GUI of an action type, then the execution script of that action type can access the value of an action instance by calling ScriptVars.Values['edFileName.Text'].
- **Self** : TForm
  This identifier points to the form of a script. You can access all the components of the form directly (ie. *edFileName.Text:='c:\test.txt';*), but the properties and methods of the form itself have to be accessed via this identifier (ie. *Self.Caption:='Form title';*).
- **StdOut** : string
  A standard output buffer, that is used as an action report. You can modify it directly, but we recommend to use a standard input/output buffer methods (read, readln, write, writeln, cls).

Special procedures and functions

- **function ActionEnabled(Sender:TObject):Boolean;**
  A common method to check if an action (Sender) is enabled
- **function ActionExecute(Sender:TObject):Boolean;**
  A common method to execute an action (Sender)
- **function ActionUpdate(Sender:TObject):Boolean;**
  A common method to perform an Update method of an action (Sender)
- **function BrowseForFolder(const Title:string; InitialDirectory: string; var SelectedDirectory:string;Position:TPoint):boolean;**
  This function invokes a dialog for selecting a folder (directory)
- **procedure Cls;**
  This procedure clears the StdOut buffer of an action
- **function CopyFileAs(const Source:string;const Target:string;FailIfExists:Boolean):Boolean;**
  Copies *Source* file to the *Target* file location
- **procedure CopyFiles(SourceDir,Mask,DestDir:string;FailIfExists:Boolean);**
  Copies all files from the *SourceDir* that match the *Mask*, to the *DestDir*
- **function DateTimeToDEF_DateTimeFormat(DT:TDateTime):string;**
  Converts a datetime to the generalized date time format (*'yyyy-mm-ddThh:mm:ss'*)
- **procedure DecodeFile(const InFile:AnsiString;const OutFile:AnsiString; const Password:AnsiString; const cMethod:AnsiString);**
  Decodes a previously encoded *InFile* by using a passed *Password* and *cMethod* (algorithm, Blowfish by default). A result is written to the *OutFile*.
  See the *EncodeFile* procedure for a list of available algorithms.
- **function DecodeText(const Text:AnsiString; const Password:AnsiString; const cMethod:AnsiString):AnsiString;**
  Decodes a previously encoded *Text* by using a passed *Password* and *cMethod* (algorithm, Blowfish by default).
  See the *EncodeFile* procedure for a list of available algorithms.

- **function DEF_DateTimeFormatToDateTime(StrTime:string):TDateTime;**
  Converts a string with a date time value (in generalized format *'yyyy-mm-ddThh:mm:ss'*) to a *TDateTime*.
- **function DeleteFilesByMask(Path,Mask: string): Boolean;**
  Deletes all the files from the *Path* directory that match the *Mask*
- **function DeleteTree(const ADir : string): boolean;**
  Deletes all the files and directories in an *ADir* directory tree
- **procedure EncodeFile(const InFile:AnsiString;const OutFile:AnsiString; const Password:AnsiString; const cMethod:AnsiString);**
  Encodes an *InFile* by using a passed *Password* and *cMethod* (algorithm, Blowfish by default). A result is written to the *OutFile*.
  The following methods (algorithms) are available:
  Gost,Blowfish,IDEA,SAFER,SAFER_K40,SAFER_SK40,SAFER_K64,SAFER_SK64,SAFER_K128,SAFER_SK128,
  TEA,TEAN,SCOP,Q128,3Way,Twofish,Shark,Square,Cast128,Cast256,Mars,Misty,RC2,RC4,RC5,RC6,FROG,
  Rijndael,Skipjack,1DES,2DES,3DES,2DDES,3DDES,3TDES,DESX,NewDES,Diamond2,Diamond2Lite,Sapphire
- **function EncodeText(const Text:AnsiString; const Password:AnsiString; const cMethod:AnsiString):AnsiString;**
  Encodes a *Text* by using a passed *Password* and *cMethod* (algorithm, Blowfish by default).
  See the *EncodeFile* procedure for a list of available algorithms.
- **function ExpandEnvVar(const Value: string): string;**
  Expands all the environment variables in a *Value*
- **function ExpandEnvVar2(var Value: string): Boolean;**
  Expands all the environment variables in a *Value*
- **function ExpandMacro(const Text:string):string;**
  Expands the Precision Builder Express macros in a *Text* including an environment variables (same as GetMacro)
- **function FloatToDEF_FloatStr(F:Extended):string;**
  Converts a float number into a string using the *optFloatFormat* variable.
- **procedure GetAllFileTimes(const FileName: string;var CreationTime,LastAccessTime,LastWriteTime:TDateTime);**
  Gets all the file times at once
- **function GetEnvVar(const Name: string): string;**
  Returns an environment variable value. Use a name without a '%' (percent) chars.
- **function GetFileCRC(const FileName:AnsiString):AnsiString;**
  Returns a file CRC32 hash value
- **function GetFileHash(const FileName:AnsiString; const hMethod:AnsiString):AnsiString;**
  Returns a file hash value by using a passed *hMethod* (a CRC32 by default).
  The following methods (algorithms) are available:
  MD4,MD5,RipeMD128,RipeMD160,RipeMD256,RipeMD320,SHA,SHA1,Haval256,Haval224,Haval192,Haval160,Haval128,
  Snefru,Tiger,Sapphire320,Sapphire288,Sapphire256,Sapphire224,Sapphire192,Sapphire160,Sapphire128,Square, XOR16,XOR32,CRC16_CCITT,CRC16_Standard,CRC32
- **function GetFileMD5(const FileName:AnsiString):AnsiString;**
  Returns a file MD5 hash value
- **procedure GetFilesByMask(Dir:string;const R:TStrings;Mask:string);**
  Fills the *R* object with a file names (including the full path) from a *Dir* that match the *Mask*
- **function GetFilesByMaskTree(Dir:string;R:TStrings;Mask:string):Boolean;**
  Fills the *R* object with a file names (including the full path) from a *Dir* (and all subdirs) that match the *Mask*
- **function GetMacro(const Text:string):string;**
  Expands the Precision Builder Express macros in a *Text* including an environment variables (same as ExpandMacro)
- **function GetSubDirsByMask(Dir:string;R:TStrings;Mask:string):Boolean;**
  Fills the *R* object with a subdirectory names (including the full path) from a *Dir* that match the *Mask*
- **function GetSubDirsByMaskTree(Dir:string;R:TStrings;Mask:string):Boolean;**
  Fills the *R* object with a subdirectory names (including the full path) from a *Dir* (and all subdirs) that match the *Mask*
- **function GetTextCRC(const Text:AnsiString):AnsiString;**
  Returns a CRC32 hash value of a *Text*
- **function GetTextHash(const Text:AnsiString; const hMethod:AnsiString):AnsiString;**
  Returns a hash value of *Text* by using a passed *hMethod*. See the GetFileHash procedure for a list of available algorithms (methods).
- **function GetTextMD5(const Text:AnsiString):AnsiString;**
  Returns a MD5 hash value of a *Text*
- **function GetVariable(const aName: string):string;**
  Returns a value of the user variable, that is defined in a current Precision Builder Express project. It looks for a value in the ScriptVars object first, then in the ProjectVars.
- **function InputQueryLM(const ACaption, APrompt: string; var Value: string):Boolean;**
  Displays a localized variant of an InputQuery dialog
- **function InputQueryLMList(const ACaption, APrompt: string; var Value: string; Items:TStrings; FixedList:boolean):Boolean;**
  Displays an enhanced localized variant of an InputQuery dialog, that supports a list of available values to choose from
- **function LoadAnsiStrFromFile(var AString: AnsiString; const FileName: string): Boolean;**
  Loads a string from a file as is (does not handle the encoding of a file)
- **function LoadStrFromFile(var AString: string; const FileName: string): Boolean;**
  Loads a string from a file with a support of transformation from various encodings (utf-8, unicode, utf-7, etc.)
- **function MessageDlgLM(const MsgCaption, Msg: string; DlgType: TMsgDlgType; Buttons:TMsgDlgButtons; HelpCtx:Longint): Integer;**
  Displays a localized variant of message dialog
- **function MimeDecodeString(const S: String): String;**
  Decodes a string *S* previously encoded with a base64 MIME encoding
- **function MimeEncodeString(const S: String): String;**
  Encodes a string *S* to the base64 MIME encoding
- **procedure Read(var s: string);**
  This procedure prompts a user for a value. A simple input box dialog is used.
- **function ReadGlobalProperty(PropertyPath:string):Variant;**

Reads a property value in a global context. The *PropertyPath* must contain a form identifier, a component identifier and a property name. It can be passed as follows: *ReadGlobalProperty('frmPrecisionBuilderE.acNew.Caption');*.

- **procedure ReadIn(var s: string);**
  This procedure prompts a user for a value. A simple input box dialog is used.
- **function ReadProperty(const Root:TComponent;const PropertyPath:string):Variant;**
  Reads a property value. *Root* is usually an identifier of an owner form,*PropertyPath* must contain a component identifier and a property name. It can be passed as follows:
  ReadProperty(frmPrecisionBuilderE,'acNew.Caption');.
- **function ReadPropertyStr(const Root:TComponent;const PropertyPath:string):string;**
  Same as *ReadProperty* above, but returns a string representation of a value.
- **function SaveAnsiStrToFile(const AString:AnsiString; const FileName:string):Boolean;**
  Saves *AString* to a file (does not save a possible UTF-8, UTF-16 preamble)
- **function SaveStrToFile(const AString:string; const FileName:string):Boolean;**
  Saves *AString* to a file with a support of writing a possible UTF-8 or UTF-16 preamble, when the string is in UTF-8 or unicode format
- **function SaveStrToFileEx(const AString:string; const FileName:string; const Encoding:string):Boolean;**
  Saves *AString* to a file using the specified encoding. You can use any known encoding identifiers, as for example (utf-8, unicode, windows-1251, etc.).
- **function SelectFile(var aFileName:string;const aFilter:string):Boolean;**
  Prompts a user for a file. *aFileName* is an initial file name,*aFilter* must be passed in a Delphi standard file format string (ie. 'Text files (*.txt,*.csv)|*.txt;*.csv|All files (*.*)|*.*').
- **function SelectFiles(var aFileNames:TStringList;const aFilter:string):Boolean;**
  Same as *SelectFile* above, but with a support of multiple files selection. A list of files is returned in*aFileNames* object.
- **function SelectFileToSave(var aFileName:string;const aFilter:string):Boolean;**
  Same as *SelectFile* above, but this variant invokes a dialog with a "save file" style.
- **function SetEnvVar(const Name, Value: string): Boolean;**
  Sets an environment variable in a context of currently logged user account
- **procedure SetGlobalEnvVar(VariableName, VariableContent: string; ForAllUsers:Boolean; BroadcastChangeMsg:boolean);**
  Sets an environment variable with a possible support of defining it for all users on a PC.
- **procedure SetVariable(const aName:string; const aValue:string);**
  Sets a value for the user defined variable of a currently opened Precision Builder Express project. If variable does not exists, it creates a new one.
- **function ShellExec(const FileName: string; const Parameters: string; const Verb: string; CmdShow: Integer): Boolean;**
  Executes an application (or opens a file) by using a standard ShellExecute API function
- **function ShellExecAndWait(const FileName: string; const Parameters: string; const Verb: string; CmdShow: Integer): Boolean;**
  Executes an application (or opens a file) by using a standard ShellExecute API function. It waits while the executed application terminates.
- **function ShellExecAndWaitWithExitCode(const FileName: string; const Parameters: string; const Verb: string; CmdShow: Integer; var ExitCode:Cardinal): Boolean;**
  Executes an application (or opens a file) by using a standard ShellExecute API function. It waits while the executed application terminates and returns its *ExitCode*.
- **function StrReplace(const S, OldPattern, NewPattern: string; All:Boolean; IgnoreCase:Boolean):string;**
  A replacement for a standard **StringReplace** Delphi function
- **function StrToDateTimeGeneral(S:string;Def:Extended):TDateTime;**
  Returns a TDateTime value of a string by trying to convert it from current locale date time format and generalized date time format. If the string does not contain a date time value, then the *Def* value is returned.
- **function StrToFloatGeneral(S:String;Def:Extended):Extended;**
  Returns a float value of a string by trying to convert it from a current locale number format and from the generalized float format (without the thousands separators). If the string does not contain a float value, then the *Def* value is returned.
- **procedure Write(s: string);**
  This procedure writes a string *s* to the StdOut buffer of an action
- **procedure WriteGlobalProperty(PropertyPath:string;Value:Variant);**
  Writes a value to the property in a global context. The *PropertyPath* must contain a form identifier, a component identifier and a property name. It can be passed as follows:
  WriteGlobalProperty('frmPrecisionBuilderE.acNew.Caption',25);.
- **procedure Writeln(s: string);**
  This procedure writes a string *s* followed by the *crlf sequence* to the StdOut buffer of an action
- **procedure WriteProperty(const Root:TComponent;const PropertyPath:string;Value:Variant);**
  Writes a value to the property. *Root* is usually an identifier of an owner form,*PropertyPath* must contain a component identifier and a property name. It can be passed as follows:
  WriteProperty(frmPrecisionBuilderE,'acNew.Caption','New file');.
- **procedure WritePropertyStr(const Root:TComponent;const PropertyPath:string;const Value:string);**
  Same as *WriteProperty* above, but the value can be passed as a string

Other identifiers

- optDateFormat : string = ShortDateFormat;
- optTimeFormat : string = 'H:mm';
- optDateTimeFormat : string = optDateFormat + ' ' + optTimeFormat;
- optFloatFormat : string = '#0.00';