What's new in EhLib.VCL 9.4

(New features of your applications)

Brief description of new features in this version:

* New type of data display in TPlannerEh - annual period broken down by days.
* In DBGridEh added the ability to save filters through the SettingsKeeper specified in the STFilter section.
* Xlsx File Generator.
* Support for exporting DBGridEh and DBVertGridEh to Xlsx files in Lazarus in Win32/64 mode and CrossPlatform.

# ***New type of data display in TPlannerEh - annual period broken down by days.***

A TPlannerHorzYearViewEh view has been added to the list of calendar views for TPlannerEh – annual period view broken down by days.



This type of calendar is used to display the full calendar year.

The calendar contains 12 rows, one row for each month.

Horizontally, the line is divided into the number of days in the month.

Columns are grouped by 7 days for each week of the month.

Holidays and weekends are highlighted in gray.

# ***In DBGridEh added the ability to save filters through the SettingsKeeper specified in the STFilter section.***

The storage technology SettingsKeeper for DBGridEh added the ability to save the filters specified in the Column.STFilter section.

The grid writes filters to the Keeper if Grid.STFilter.Visible = True when calling the TCustomDBGridEh.WriteSettings (Keeper: TSettingsKeeperEh) function without specifying additional parameters GridSettingsKeeperOptions and ColumnSettingsKeeperOptions

See an example of using SettingsKeeper in Demos\JSonSettingsKeeper\JSonSettingsKeeper.dpr.

# ***Xlsx File Generator.***

A new class has been added to the library - TXlsMemFileEh, which allows you to create Xlsx files without the MSOffice been installed.

TXlsMemFileEh is not a visual component. It is not listed in the IDE component palette. The class must be used directly in the program code. The class is described in the XlsMemFilesEh module.

With this class, you can quickly generate files in MSExcel format. The created files can also be opened using the OpenOffice Calc application.

In the current version, TXlsMemFileEh supports the following features when generating Xlsx files:

Working with a class involves the following usage scenario:

- Creating an instance of the class TXlsMemFileEh

var

 xlsFile: TXlsMemFileEh;

begin

 xlsFile := TXlsMemFileEh.Create;

- When creating an instance, a single Worksheet object is created. You can customize the properties of an existing sheet and / or add new ones.

xlsFile.Workbook.Worksheets[0].Name := 'DBGrid';

xlsFile.Workbook.AddWorksheet('VertGrid');

- Specify the width of the columns in the worksheet.

 Sheet.Columns[0].Width := Sheet.Columns.ScreenToXlsWidth(DBGridEh1.Columns[0].Width);

 Sheet.Columns[1].Width := Sheet.Columns.ScreenToXlsWidth(DBGridEh1.Columns[1].Width);

- Fill in the contents of the worksheets.

 Sheet.Cells[0, 3].Value := DBGridEh1.Columns[0].Title.Caption;

 Sheet.Cells[1, 3].Value := DBGridEh1.Columns[1].Title.Caption;

 Sheet.Cells[2, 3].Value := DBGridEh1.Columns[2].Title.Caption;

…

 Sheet.Cells[0, i + 4].Value := DBGridEh1.Columns[0].Field.Value;

 Sheet.Cells[1, i + 4].Value := DBGridEh1.Columns[1].Field.Value;

 Sheet.Cells[2, i + 4].Value := DBGridEh1.Columns[2].Field.Value;

 Sheet.Cells[1, i+4].Formula := 'SUM(E5:' + 'E' + IntToStr(i+4) + ')';

Filling cell values is supported Cells[ColIndex, RowIndex].Value and formulas Cells[ColIndex, RowIndex].Formula.

- Customize cell format and cell borders on a sheet.

 cr := Sheet.GetCellsRange(0,i+4,6,i+4);

 cr.Border.Top.Style := clsMediumEh;

 cr.Border.Bottom.Style := clsMediumEh;

 cr.Border.Left.Style := clsMediumEh;

 cr.Border.Right.Style := clsMediumEh;

 cr.Font.IsBold := True;

 cr.ApplyChages;

To change the format of a cell or group of cells, you must first obtain a group of cells using the GetCellsRange method. Then in the resulting class, change the properties to set the format. The GetCellsRange method returns the type of the interface. After using the CellsRange instance, it is not necessary to delete it, it will be deleted automatically when the last pointer to it goes out of scope.

The IXlsFileCellsRangeEh interface has the following properties for setting the format of a range of cells:

Фонт (Font)

Background shading (Fill)

Setting the outer edges of the selected range (Border.Left, Border.Right, Border.Top, Border.Bottom).

Adjustment of internal dividing lines in the selected range (InsideBorder)

Horizontal and vertical text alignment (HorzAlign и VertAlign)

Word wrap (WrapText)

Rotate text (Rotation)

Number format - NumberFormat (Described in format of MSExcel).

- Save created file to disk.

 xlsFile.SaveToFile(Path);

- Delete TXlsMemFileEh object.

 xlsFile.Free;

You can also use additional properties and methods to implement the following features when creating Xlsx files:

- Merging cells:

Sheet.MergeCell(0,0, 6,0);

- Examples of setting the format for a range of cells:

 cr := Sheet.GetCellsRange(0,4,6,i+4);

 cr.Border.Top.Style := clsMediumEh;

 cr.Border.Bottom.Style := clsMediumEh;

 cr.Border.Left.Style := clsMediumEh;

 cr.Border.Right.Style := clsMediumEh;

 cr.InsideBorder.Top.Style := clsThinEh;

 cr.InsideBorder.Bottom.Style := clsThinEh;

 cr.InsideBorder.Left.Style := clsThinEh;

 cr.InsideBorder.Right.Style := clsThinEh;

 cr.NumberFormat := '#,##0.0000';

 cr.VertAlign := cvaCenterEh;

 cr.HorzAlign := chaCenterEh;

 AFont := DBVertGridEh1.VisibleFieldRow[i].Font;

 cr.Font.Name := AFont.Name;

 cr.Font.Size := AFont.Size;

 cr.Font.Color := AFont.Color;

 cr.Font.IsBold := fsBold **in** AFont.Style;

 cr.Font.IsItalic := fsItalic **in** AFont.Style;

 cr.Font.IsUnderline := fsUnderline **in** AFont.Style;

 cr.ApplyChages;

- Specifying frozen columns and rows:

 Sheet.FrozenRowCount := 4;

 Sheet.FrozenColCount := 3;

- Setting the auto-filter area:

 Sheet.AutoFilterRange.FromCol := 0;

 Sheet.AutoFilterRange.FromRow := 3;

 Sheet.AutoFilterRange.ToCol := 6;

 Sheet.AutoFilterRange.ToRow := i+4-1;

The full version of the demo project demonstrating all the features of the class TXlsMemFileEh, see the folder <EhLib archive>\Demos\XlsFileEh\

The following features are not supported in the current version:

* The ability to create files in binary format Xls is not supported.
* The ability to load files is not supported. Files can be created only from scratch in the program code.

Restrictions:

* Full-featured work with the TXlsMemFileEh class is possible in Delphi version XE2 and higher or in Lazarus. To work with the TXlsMemFileEh class in earlier versions of Delphi, you need to use third-party functions to generate Zip files. Read more about creating Zip file generation for Delphi version below XE2 in the section files <EhLib Archive>\ZipProviders\.

# ***Support for exporting DBGridEh and DBVertGridEh to Xlsx files in Lazarus in Win32/64 mode and CrossPlatform.***

Starting with this version, support for exporting DBGridEh and DBVertGridEh to Xlsx files in Lazarus in Win32 / 64 and CrossPlatform mode has been added to the library.

In Lazarus, the ExportDBGridEhToXlsx and ExportDBVertGridEhToXlsx functions work.

The generated Xlsx files are compatible with the tabular data editing program MS Excel and OpenOffice Calc.