Новые возможности EhLib.VCL 9.5

(Новые возможности ваших приложений)

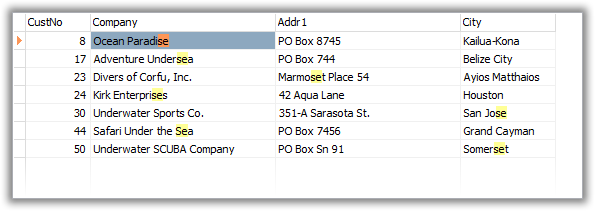
Краткое описание новых возможностей этой версии:

* SearchPanel - режим поиска и фильтрации по месту
* Расширенная фильтрация выпадающих списков в компонентах TDBComboboxEh и TDBLookupComboboxEh
* Другие изменения и дополнения

# ***SearchPanel - режим поиска и фильтрации по месту***

В SearchPanel добавлен новый режим поиска и фильтрации – поиск и фильтрация по месту (SearchPanel.Location = splCellInplaceEh).

В данном режиме панель поиска не отображается над гридом. Поиск включается по комбинации клавиш Ctrl+F. Текст поиска и фильтрации вводится непосредственно в ячейке грида. По мере ввода текста грид сразу выполняет поиск первого найденного фрагмента и, в случае установленного свойства FilterOnTyping, выполняется фильтрация для отображения только найденных строк.



Данный режим поиска является расширенной версией поиска OptionsEh.dghIncSearch.

Для включения поиска и фильтрации по месту:

Установите свойство SearchPanel.Enabled = True

Установите свойство SearchPanel.FilterOnTyping = True

Для включения автоматической фильтрации установите свойство SearchPanel.FilterOnTyping = True.

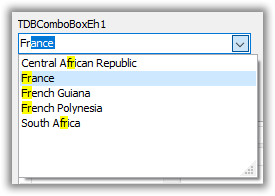
Если установить свойство SearchPanel.PreferSearchToEdit в True, то при нажатии на буквенно-цифровые клавиши Грид будет сразу входить в режим поиска, а не открывать редактор текста ячейки. Текстовый редактор может быть открыт по F2 или Enter.

В run-time для включения режиме поиска нажмите комбинацию клавиш - Ctrl+F.   
Для перехода к следующей/предыдущей найденной ячейки используйте клавиши Вниз/Вверх.  
Для выхода из режима поиска используйте клавиши ESCAPE, ENTER или F2.

# ***Расширенная фильтрация выпадающих списков в компонентах TDBComboboxEh и TDBLookupComboboxEh***

**Возможность фильтрации выпадающего списка при вводе текста в компоненте В TDBComboboxEh**

В TDBComboboxEh добавлено возможность фильтрации выпадающего списка при вводе текста. Аналогичная фильтрация уже присутствует в компоненте TDBLookupComboboxEh.



В TDBComboboxEh.DropDownBox добавлены свойства:

**property** AutoFilter: Boolean;

Определяет, активна ли фильтрация данных при вводе текста в редакторе TDBComboboxEh.

**property** AutoFilterType: TLSAutoFilterTypeEh;

Задает тип фильтрации. lsftBeginsWithEh – фильтрация по началу текста ячейки, lsftContainsEh – фильтрация с любой позиции в тексте ячейки.

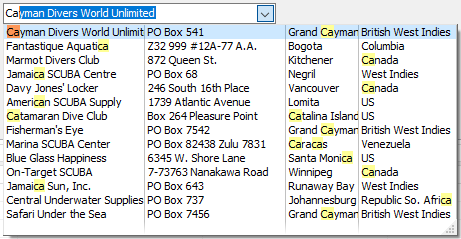
Фильтрация работает в компоненте TDBComboboxEh, а также в столбцах грида DBGridEh когда заполнены свойства PickList или PickList/KeyList.

Для настройки фильтра в столбцах грида используйте свойства TColumnEh.DropDownBox.ListSourceAutoFilter и TColumnEh.DropDownBox.ListSourceAutoFilterType.

**Расширенная фильтрация выпадающего списка при вводе текста в компоненте** **TDBLookupComboboxEh**

В TDBLookupComboboxEh добавлено свойство TDBLookupComboboxEh.DropDownBox.ListSourceAutoFilterAllColumns типа Boolean.

Свойство позволяет фильтровать данные по всем столбцам выпадающего списка.



Список полей обычно задается свойством TDBLookupComboboxEh.ListField

ListField = 'Company;Addr1;City;Country'

Также может понадобиться возможность вводить текст в TDBLookupComboboxEh без ограничения на вводимые значения только из списка значений в поле DisplayField. Для этого установите свойство Style = csDropDownEh.

# ***Другие изменения и дополнения.***

* **Событие TPrintDBGridEh.OnPrintDataCell**

В TPrintDBGridEh добавлено событие OnPrintDataCell. Событие можно использовать для дополнительной отрисовки содержимого ячейки при печати грида.

В следующем примере обработчик события рисует значок ProgressBar при печати ячейки поля 'Population'.

**procedure** TForm1.PrintDBGridEh1PrintDataCell(Sender: TPrintDBGridEh;

VPrinter: TVirtualPrinter; Column: TColumnEh; **const** ARect: TRect;

**var** Params: TPrintColCellParamsEh; **var** Processed: Boolean);

**begin**

**if** (Column.FieldName = 'ProgressBar') **then**

**begin**

DrawProgressBarEh(MemTableEh1.FieldByName('Population').AsFloat, 0, 50000000,

VPrinter.Canvas, ARect, clLtGray, cl3DDkShadow, clNone);

Processed := True;

**end**;

**end**;

* **Событие TMemTableEh.OnRecordsViewTreeNodeCollapsed**

В TMemTableEh добавлено событие OnRecordsViewTreeNodeCollapsed. Событие срабатывает при свертывании ветки дерева в режиме TreeVIew.

* **Кодировка при экспорте с помощью TDBGridEhExportAsText.**

В TDBGridEhExportAsText добавлены свойства WriteBOM и Encoding.

**property** WriteBOM: Boolean;

Задают необходимость записи признака Маркер последовательности байтов (BOM) при экспорте в поток.

**property** Encoding: TEncoding;

Задает кодировку текстовой информации при импорте грида в поток.

В следующем примере показан код для экспорта содержимого DBGridEh1 в файл в формате UTF8 и записью маркера BOM.

**procedure** TForm1.Button1Click(Sender: TObject);

**var**

Path: String;

gridExp: TDBGridEhExportAsText;

**begin**

GetDir(0, Path);

Path := Path + '\ExpFile.Txt';

gridExp := TDBGridEhExportAsText.Create;

gridExp.DBGridEh := DBGridEh1;

gridExp.Encoding := TEncoding.UTF8;

gridExp.WriteBOM := True;

**try**

gridExp.ExportToFile(Path, True);

**finally**

gridExp.Free;

**end**;

**end**;

* **Тонкая настройка формирования строки фильтра DBGridEh.STFilter для DataSet’а через компонент TDatasetFeaturesEh.**

TDatasetFeaturesEh добавлены виртуальные методы для тонкой настройки при формировании строки фильтрации на основе фильтра, установленного в DBGridEh.STFilter.

Для формирования и применения фильтра из DBGridEh.STFilter в DataSet используется метод ApplyFilter. В предыдущих версиях библиотеки метод ApplyFilter вызывал глобальные функции для формирования строки фильтра и применения фильтра. Однако глобальные функции не учитывали некоторые особенности некоторых DataSet’ов.

В текущей версии в TDatasetFeaturesEh добавлены следующие виртуальные функции, которые позволяют более тонко настроить строку фильтра и метод применения фильтра к DataSet’у:

**procedure** ApplyGridLocalFilter(Grid: TCustomDBGridEh; DataSet: TDataSet; IsReopen: Boolean); virtual;

Метод вызывается, когда необходимо установить локальный фильтра в DataSet (DBGridEh.STFilter.Local = True). Обычно при локальной фильтрации строка фильтра присваивается свойству DataSet.Filter.

**function** GetGridFilterAslFilterString(Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод формирует и возвращает строку фильтрации на основе фильтра в DBGridEh. Фильтр храниться в свойства столбца TColumnEh.STFilter.

Свойство IsLocalFilter определяет, что надо формировать выражение для локальной фильтрации.

Метод должен пройтись по столбцам грида и вызвать GetColumnExpressionAsFilterString для каждого столбца.

**function** GetColumnExpressionAsFilterString(Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод формирует и возвращает строку фильтрации для одного столбца TColumnEh.

**function** GetBinaryExpressionAsFilterString(FieldName: String; AnOperator: TSTFilterOperatorEh; AnOperand: Variant; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод формирует и возвращает строку фильтрации для одного столбца TColumnEh на основе значения параметров FieldName, AnOperator, AnOperand.

**function** GetColumnFilterFieldName(Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает правильное имя поля, которое будет участвовать в формировании выражения фильтрации.

**function** ColumnOperatorValueToFilterStrValue(AnOperator: TSTFilterOperatorEh; AnOperand: Variant; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое выражение на основе значения полей AnOperator и AnOperand.

**function** IsInOperatorSupported(Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): Boolean; virtual;

Метод возвращает True если DataSet и метод фильтрации поддерживает оператор In.

**function** IsLikeOperatorSupported(Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): Boolean; virtual;

Метод возвращает True если DataSet и метод фильтрации поддерживает оператор Like.

**function** IsFilterUseFieldOrigin(Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): Boolean; virtual;

Метод возвращает True если в качестве названия поля надо использовать Field.Origin.

**function** GetNullComparisionFilterString(AnOperator: TSTFilterOperatorEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое выражения для сравнения поля с Null. IS NULL или = Null. IS NOT NULL или <> Null

**function** GetOperatorFilterStrValue(AnOperator: TSTFilterOperatorEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое представления оператора сравнения, заданного параметров AnOperator.

**function** FilterFieldNameToStrValue(FieldName: String; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое значение имени поля. Некоторые DataSet’ы при локальной фильтрации требуют, чтобы поле заключалось в квадратные скобки – ‘[FieldName]’.

**function** ExpressionValueToFilterStrValue(AnOperator: TSTFilterOperatorEh; Value: Variant; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое значение выражения для раздела выражения Value.

**function** DateTimeValueToFilterStrValue(AnOperator: TSTFilterOperatorEh; Value: TDateTime; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое значение выражения для константы с типом DataTime.

**function** FloatValueToFilterStrValue(AnOperator: TSTFilterOperatorEh; Value: Extended; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое значение выражения для константы с типом Float.

**function** VarValueToFilterStrValue(AnOperator: TSTFilterOperatorEh; Value: Variant; Column: TColumnEh; Grid: TCustomDBGridEh; DataSet: TDataSet; IsLocalFilter: Boolean): String; virtual;

Метод возвращает текстовое значение выражения для константы с типом String.

**procedure** ApplyGridServerFilter(Grid: TCustomDBGridEh; DataSet: TDataSet; IsReopen: Boolean); virtual;

Метод вызывается, когда необходимо установить серверный фильтр в DataSet (DBGridEh.STFilter.Local = False).