



Руководство разработчика FastCube .NET

Версия 2021.1

© 2008-2021 ООО Фаст Репортс

Компоненты FastCube .NET

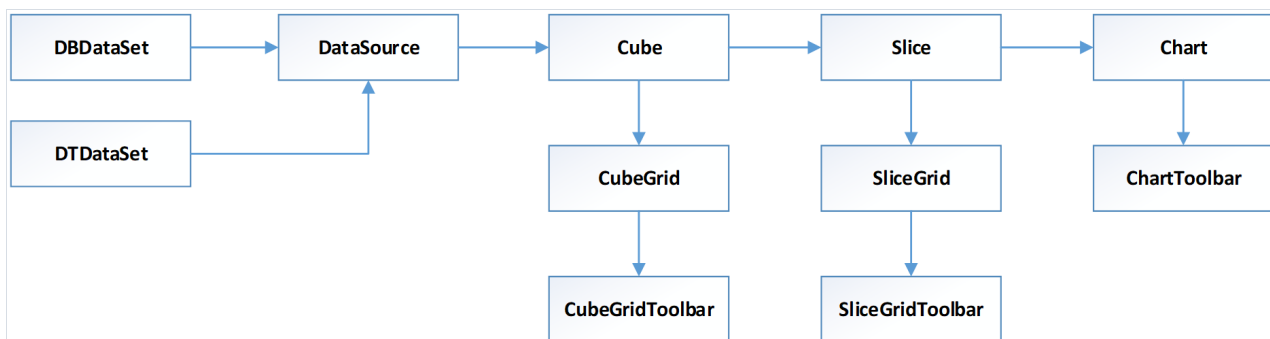
FastCube .NET представляет собой библиотеку компонентов для Visual Studio .NET. Компоненты можно разделить на визуальные и не визуальные.

Данное руководство призвано помочь вам создавать и редактировать куб, срез и другие компоненты и их свойства в коде приложения.

Список компонентов FastCube .NET:

- **Cube** – невидуальный компонент, отвечающий за загрузку полей, уникальных значений и строк данных из файлов, баз данных и других источников, хранение и обработку структур в памяти приложения.
- **CubeGrid** – визуальный компонент, отображающий данные **Cube** в табличном виде с возможностью манипуляций со стороны пользователя.
- **CubeGridToolbar** – визуальный компонент, панель инструментов с функциями управления **CubeGrid**.
- **Slice** – невидуальный компонент, отвечающий за выполнение OLAP-операций над данными **Cube**, расчет показателей и подготовку представлений для сводной таблицы и диаграммы.
- **SliceGrid** – визуальный компонент, отображающий данные **Slice** в виде сводной таблицы, с возможностью манипуляций со стороны пользователя.
- **SliceGridToolbar** – визуальный компонент, панель инструментов с функциями управления для **SliceGrid**.
- **Chart** – визуальный компонент, отображающий данные **Slice** в виде диаграммы.
- **ChartToolbar** – визуальный компонент, панель инструментов с функциями управления для **Chart**.
- **DataSource** – невидуальный компонент, источник данных для куба.
- **DBDataSet** – невидуальный компонент, поставляющий данные в **DataSource** из базы данных.
- **DTDataSet** – невидуальный компонент, поставляющий данные в **DataSource** из **DataTable**.

Теперь, давайте рассмотрим схему взаимосвязи компонентов.



Рассматриваем схему слева направо. Объект **DataSource** имеет свойство **DataSet**. Значением этого свойства будет ссылка на один из двух объектов: **DBDataSet** или **DTDataSet**.

Объект **Cube** имеет свойство **DataSource**. А **CubeGrid** и **Slice** связаны с объектом **Cube**.

В свою очередь **SliceGrid** и **Chart** связаны с объектом **Slice**. В дальнейшем, эта схема поможет нам в настройке связи компонентов.

Cube

Компонент **Cube** – это основа всего FastCube. Он хранит данные куба и выступает поставщиком данных и схемы данных для других компонентов.

Данные в куб могут быть загружены из базы данных, из источника пользователи, либо из сохранённого куба.

Ниже будут приведены основные свойства и методы этого компонента.

Основные свойства

Свойство	Описание
public bool Active	Активность куба (загруженность данными) true или false. После открытия куба свойство Active имеет значение true. Только в этом случае визуальные компоненты отображают данные.
public string Caption	Заголовок куба. Виден в CubeGrid.
public bool CompressCubeFile	Использовать ли компрессию при сохранении данных куба для последующего использования.
public string Description	Описание.
public bool SkipFieldsWithErrors	Указывает пропускать ли поля источника данных с ошибками (неподдерживаемый тип и т.п.) или выводить Exception
public SourceType SourceType	Тип источника данных указывает откуда брать данные для заполнения куба. Может принимать одно из значений: <code>Empty</code> , <code>DataSource</code> , <code>File</code> , <code>Stream</code> , <code>Manual</code>

Основные методы

Метод	Описание
public void ClearGroups()	Отчистить группировки. При этом все применённые группировки будут очищены.
public void Close()	Закреть куб. При этом данные куба и его структуры будут очищены.
public int GetFieldsCount()	Получить количество полей в кубе.
public int GetSourceRecordsCount()	Получить количество исходных записей данных в кубе.
public Utils.Variant GetSourceValue(int recordIndex, int fieldIndex)	Получить значение поля источника данных в указанной записи. Параметры – индекс записи и индекс поля.
public Utils.Variant GetSourceValue(int recordIndex, CubeField cubeField)	Получить значение поля источника данных в указанной записи. Параметры – индекс записи и поле куба.

Метод	Описание
public string GetSourceValueAsString(int recordIndex, int fieldIndex)	Получить значение поля источника данных в указанной записи в виде строки. Параметры – индекс записи и индекс поля.
public string GetSourceValueAsString(int recordIndex, CubeField cubeField)	Получить значение поля источника данных в указанной записи в виде строки. Параметры – индекс записи и поле куба.
public int GetSourceValueId(int recordIndex, int fieldIndex)	Получить идентификатор уникального значения поля источника данных в указанной записи. Параметры – индекс записи и индекс поля. Параметры – индекс записи и индекс поля.
public int GetSourceValueId(int recordIndex, CubeField cubeField)	Получить идентификатор уникального значения поля источника данных в указанной записи.
public int GetSourceValueIdAndVariant(int recordIndex, CubeField cubeField, ref Utils.Variant value)	Получить идентификатор уникального значения поля источника данных и само значение в указанной записи. Параметры – индекс записи, поле куба и значение.
public int GetSourceValueIdAndVariant(int recordIndex, int fieldIndex, ref Utils.Variant value)	Получить идентификатор уникального значения поля источника данных и само значение в указанной записи. Параметры – индекс записи, индекс поля и значение.
void Load(Stream stream)	Загружает сохранённые данные куба из потока. Куб очищается перед загрузкой.
void Load(string fileName)	Загружает сохранённые данные куба из указанного файла. Куб очищается перед загрузкой.
public void Open()	Открыть куб. Загрузка куба из источника данных указанного в настройках.
public void Save(Stream s, bool compress = true, object filter = null)	Записывает структуру куба и данные в поток.
public void Save(string fileName)	Записывает структуру куба и данные в указанный файл.

Загрузка куба из ранее сохранённого файла

```
FastReport.Olap.Cube.Cube cube = new FastReport.Olap.Cube.Cube();
cube.Load("C:\Program Files (x86)\FastReports\FastCube.Net
Professional\Demos\Data\Cubes\2_0_sample_en1.mdc");
```

Загрузка куба из источника данных

```
cube.SourceType = SourceType.DataSource;
cube.DataSource = DataSource1;
cube.Open();
```

CubeGrid

Компонент **CubeGrid** представляет собой таблицу, заполняемую данными из куба. Проще говоря визуальное отображение куба.

Свойства

Свойство	Описание
public Cube.Cube Cube	Объект куба.
ublic CubeDataZone DataZone	Настройки отображения данных в сетке.

Методы

Метод	Описание
public bool Export(ExportBase export)	Экспорт куба в один из форматов: HTML, DBF, CSV, XML, Open Document Spreadsheet, Excel, Excel 2007. Возвращает успех или неудачу.
public override string GetClipboardText()	Возвращает строковое представление выделенных в таблице данных куба.

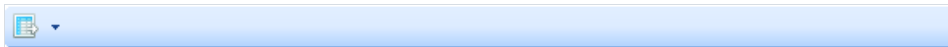
Настройка CubeGrid

```
CubeGrid cubeGrid = new CubeGrid();
cubeGrid.Dock = DockStyle.Fill;
cubeGrid.Parent = this;
cubeGrid.Cube = cube;
```

Приведенный пример, показывает, как создать cubeGrid из кода приложения. Созданный объект нужно поместить на форму (свойство Parent), и настроить расположение (свойство Dock). Кроме того, необходимо указать куб, из которого брать данные.

CubeGridToolbar

Компонент **CubeGridToolbar** – представляет собой панель инструментов, которая работает в связке с компонентом cubeGrid.



Эта панель инструментов предоставляет только один элемент – экспорт. В таблице ниже будут приведены доступные форматы экспорта.

Свойства

Свойство	Описание
<code>public CubeGrid Grid</code>	Грид, для которого действует панель инструментов.

Настройка CubeGridToolbar

```
CubeGridToolbar cubeGridToolbar = new CubeGridToolbar();
cubeGridToolbar.Dock = DockStyle.Top;
cubeGridToolbar.Parent = this;
cubeGridToolbar.Grid = cubeGrid;
```

Slice

Компонент **Slice** содержит в себе настройки среза куба и выполняет операции по группировки и вычислению данных. Управлять настройками среза можно через подключенный компонент `SliceGrid`, из кода программы, либо загрузить ранее сохраненную схему из файла или потока.

Свойства

Свойство	Описание
public bool AutoUniqueValuesFilter	Указывает применять мгновенный пересчет фильтра по уникальному значению поля или обрабатывать его в пакетном режиме. Актуально для выпадающего списка уникальных значений в SliceGrid
public Cube.Cube Cube	Объект куба.
public FieldComparerType FieldsOrder	Порядок отображения списка полей: ByIndex, ByName, ByCaption.
public int ColCount	Возвращает количество колонок в сводной таблице.
public bool HideColZeros	Скрывать пустые колонки (колонки, в которых все значения равны 0 или пустые).
public bool HideRowZeros	Скрывать пустые строки (строки, в которых все значения равны 0 или пустые).
public bool HideTotalForSingleValue	Скрывать итог если узел содержит всего одно значение.
public int RowCount	Возвращает количество строк в сводной таблице.
public Types.Language ScriptLanguage	Язык скрипта. Представляет собой перечисление со значениями: <code>CSharp = 0, Vb = 1</code>
public string ScriptText	Текст скрипта.
public PermissionSet ScriptRestrictions	Настройка ограничений скрипта.
public SliceFields SliceFields	Контейнер, который содержит все поля среза.
public AxisContainer XAxisContainer	Контейнер для полей, размещающихся по оси X. Для измерений.
public AxisContainer YAxisContainer	Контейнер для полей, размещающихся по оси Y. Для измерений.
public FiltersContainer FiltersContainer	Контейнер для полей располагаемых в зоне фильтров.

Свойство	Описание
public MeasuresContainer MeasuresContainer	Контейнер, который содержит показатели.

ОСНОВНЫЕ МЕТОДЫ

Метод	Описание
public void BeginUpdate()	Включить режим обновления (пакетного изменения схемы).
public void EndUpdate()	Завершить обновление (пакетного изменения схемы).
public void Clear()	Отчистить срез. Сброс всех настроек.
public void Save(Stream stream, SliceSaveExtras extras = SliceSaveExtras.None)	Записывает схему среза в поток. Параметр SliceSaveExtras указывает дополнительные настройки сохранения (None, Filters, Groups, Charts).
public void Save(XmlDocument doc, SliceSaveExtras extras = SliceSaveExtras.None)	Записывает схему среза в XML документ. Параметр SliceSaveExtras указывает дополнительные настройки сохранения (None, Filters, Groups, Charts).
public void Save(string fileName, SliceSaveExtras extras = SliceSaveExtras.None)	Записывает схему среза в указанный файл. Параметр SliceSaveExtras указывает дополнительные настройки сохранения (None, Filters, Groups, Charts).
public bool Load(string fileName)	Загрузить схему среза из файла с заданным именем. Схема среза, настройки групп, фильтры и диаграммы сбрасываются перед загрузкой. При успехе возвращает true.
public bool Load(XmlDocument doc)	Загрузить схему среза из XML документа. Схема среза, настройки групп, фильтры и диаграммы сбрасываются перед загрузкой. При успехе возвращает true.
public bool Load(Stream stream)	Загрузить схему среза из потока. Схема среза, настройки групп, фильтры и диаграммы сбрасываются перед загрузкой. При успехе возвращает true.
public void SetColsWidth(int value)	Установить ширину всех колонок. Значение в пикселях.
public void SetColWidth(int columnIndex, int value)	Установить ширину заданной колонки. Значение в пикселях.
public void SetRowHeight(int rowIndex, int value)	Установить высоту заданной строки. Значение в пикселях.
public void SetRowsHeight(int value)	Установить высоту всех строк. Значение в пикселях.
public void Transpose()	Транспонировать срез (поменять оси).

Настройка среза


```
FastReport.Olap.Slice.Slice slice1 = new FastReport.Olap.Slice.Slice();
slice1.Cube = cube;
```

Настройка структуры среза

Настройка структуры среза заключается в добавлении полей и показателей в нужные контейнеры.

Для добавления и удаления полей в контейнеры оси (XAxisContainer, YAxisContainer), служат следующие методы:

Метод	Описание
public int AddSliceField(SliceField sliceField)	Добавить поле в срез. Возвращает индекс добавленного элемента.
public int InsertSliceFieldToPosition(SliceField sliceField, int index)	Вставить поле среза в указанную позицию. Возвращает индекс добавленного элемента.
public void RemoveAxisField(AxisField axisField)	Удалить поле из контейнера оси.
public int AddMeasuresField()	Добавляет в ось поле "Показатели".
public void DeleteMeasuresField()	Убирает из оси поле "Показатели".

Для добавления и удаления показателей в контейнер показателей (MeasuresContainer) служат следующие методы:

Метод	Описание
public int AddMeasure(MeasureField measureField)	Добавить показатель в контейнер. Возвращает индекс добавленного элемента.
public void DeleteMeasure(MeasureField measureField)	Удалить показатель из контейнера. При этом поле переносится в область фильтров.
public void DeleteMeasures()	Удалить все показатели из контейнера.
public void DeleteMeasure(int measureIndex)	Удалить показатель из контейнера по индексу.
public void InsertMeasure(MeasureField measureField, int index)	Вставить показатель в указанную позицию.

Пример:

```

//Начать изменение структуры среза
slice1.BeginUpdate();

//Добавить поле в контейнер оси X
slice1.YAxisContainer.AddSliceField(slice1.SliceFields.GetFieldByIndex(0));

//Вставить поле SliceField1 в контейнер оси Y в 5 позицию
slice1.YAxisContainer.InsertSliceFieldToPosition(slice1.SliceFields.GetFieldByName("SliceField1"), 5);

//Добавить поле "Показатели" в контейнер оси X
slice1.XAxisContainer.AddMeasuresField();

//Добавить поле в контейнер "Показатели"
slice1.MeasuresContainer.AddMeasure(new MeasureField(slice1, FastReport.OLAP.Types.AggregateFunction.Sum,
slice1.SliceFields.GetFieldByIndex(1), null, null, "Measure1", "Measure1", false));

//Закончить обновление структуры среза
slice1.EndUpdate();

```

Показатели можно перемещать в списке для определения порядка их отображения с помощью метода `public bool MoveMeasure(int fromIndex, int toIndex)`. Параметры: индекс элемента который нужно переместить, индекс куда переместить. При успехе возвращает `true`.

Методы `SetAllVisible()` и `SetNoneVisible()` позволяют скрыть все показатели или отобразить все показатели.

Для того, чтобы скрыть или отобразить конкретный показатель, используйте конструкцию:

```

slice1.MeasuresContainer.GetMeasureFieldByIndex(1).Visible = true;

```

SliceGrid

Компонент **SliceGrid** отображает срез в виде кросс-таблицы. Это основной инструмент аналитика. Позволяет настраивать расположение полей измерений и показателей, добавлять новые, сортировать, группировать и другое.

Свойства

Свойство	Описание
public SliceDataZone DataZone	Настройки отображения данных в сетке.
public Slice.Slice Slice	Задается объект среза.
public XAxisZone XAxisZone	Настройки зоны отображения оси X.
public YAxisZone YAxisZone	Настройки зоны отображения оси Y.
public SliceItemsZone FilterFieldsZone	Настройки зоны фильтров.
public SliceItemsZone XFieldsZone	Настройки зоны отображения измерений по оси X.
public SliceItemsZone YFieldsZone	Настройки зоны отображения измерений по оси Y.
public SliceFieldsZone FieldsZone	Настройки зоны списка полей.

Методы

Метод	Описание
public bool Export(ExportBase export)	Метод экспорта отчета в один из форматов: HTML, DBF, CSV, XML, Open Document Spreadsheet, Excel, Excel 2007.
public void ShowFieldsEditor()	Открыть окно со списком полей среза.
public override string GetClipboardText()	Получить текстовое представление выделенных ячеек.

Настройка SliceGrid

```
SliceGrid sliceGrid = new SliceGrid();
sliceGrid.Dock = DockStyle.Fill;
sliceGrid.Parent = this;
sliceGrid.Slice = slice1;
```

Этот пример показывает, как создать объект sliceGrid в коде приложения. Если такой необходимости нет, то все настройки можно произвести в инспекторе свойств.

SliceGridToolbar

Компонент **SliceGridToolbar** представляет собой визуализированную панель инструментов для SliceGrid:



Состав инструментария следующий:

1. Сохранить:
 - Куб;
 - Схему.
2. Открыть:
 - Куб;
 - Дополнительный куб (дополнение данных в активный куб);
 - Схему.
3. Отчистить настройки;
4. Экспортировать в:
 - HTML;
 - DBF;
 - CSV;
 - XML;
 - Open Document Spreadsheet;
 - Excel;
 - Excel 2007.
5. Транспонировать – поменять оси X и Y местами;
6. Скрыть пустые строки;
7. Скрыть пустые столбцы;
8. Тип сортировки строк:
 - Сортировать по значению оси;
 - Сортировать по итогам показателей;
 - Сортировать по выделенной колонке.
9. Тип сортировки колонок:
 - Сортировать по значению оси;
 - Сортировать по итогам показателей;
 - Сортировать по выделенной строке.
10. Редактировать показатели – настройки показателей. Сюда входит настройка условного выделения данных;
11. Формат отображения данных – задается для выбранной колонки или строки;
12. Список полей – список всех полей, доступных в срезе;
13. Редактор формул – по сути редактор скрипта на языке программирования C# или VB;
14. Информация – информация о срезе.

Свойства

Свойство	Описание
public SliceGrid Grid	Грид, для которого действует панель инструментов

Свойство**Описание**

public override List<MetalItem> ToolItems	Список элементов панели инструментов
public string DialogsDefaultPath	Путь по умолчанию для диалогов открытия/сохранения

Настройка SliceGridToolbar

```
FastReport.Olap.Controls.SliceGridToolbar toolbar = new FastReport.Olap.Controls.SliceGridToolbar();  
toolbar.Grid = sliceGrid1;  
toolbar.Parent = sliceGrid1;  
toolbar.Dock = DockStyle.Top;
```

Chart

Компонент **Chart** представляет собой график, основанный на данных из среза. Он строится автоматически, достаточно лишь задать свойство `Slice`.

Свойства

Свойство	Описание
public MarksShowStyle MarksShowStyle	Стиль отображения меток
public SeriesType SeriesType	Тип Series (колончатая, круговая и др.)
public bool SkipNullPoints	Пропускать пустые точки на диаграмме
public AxisDataType BaseAxisDataType	Тип данных базовой оси
public int MeasureFieldIndex	Индекс поля показателя
public int SeriesFieldCount	Количество полей, используемых в качестве источника для Series
public int CategoriesFieldCount	Количество полей, используемых в качестве источника для категорий
public AxisRegion SeriesAxis	Ось – источник Series.
public AxisRegion CategoriesAxis	Ось – источник категорий.
public ChartDataType DataType	Способ получения данных в диаграмме: VuAxisAxis – категории и Series из измерений. Используется для отображения одного показателя; VuAxisMeasures – категории из измерений, Series из показателей; VuMeasuresAxis – категории из показателей, Series из измерений;
public Slice.Slice Slice	Ссылка на объект slice
public bool Frozen	Заморозка позволяет зафиксировать состояние графика, так, что последующие изменения в срезе не будут на нем отражаться

Методы

Метод	Описание
public void BeginUpdate()	Включить режим пакетного редактирования

Метод	Описание
public void EndUpdate()	Завершить пакетного редактирование
public void Load(XmlItem item)	Загрузить настройки графика из XML элемента
public void Save(XmlItem item)	Сохранить настройки графика в XML элемента

Настройка Chart

Настройка компонента Chart сводится к выбору доступного среза в свойстве Slice.

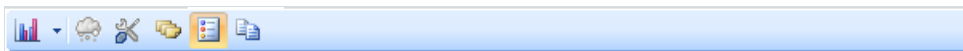
Чтобы создать и настроить объект Chart в коде приложения используйте код:

```
Chart chart = new Chart();  
chart.Dock = DockStyle.Fill;  
chart.Parent = tabPage3;  
chart.Slice = slice1;
```

В этом случае нам нужно создать объект, настроить его отображение, привязать к родительскому объекту и задать срез.

ChartToolbar

Компонент **ChartToolbar** содержит инструменты для настройки отображения диаграммы.



Состав:

1. Chart style – стиль диаграммы:
 - Bar - столбцы;
 - Line - линейная;
 - Point - точки;
 - Area – цветные области;
 - Pie - круговая;
 - Horiz Bar – горизонтальные полосы.
2. Frozen chart – заморозить текущее состояние диаграммы;
3. Chart properties – свойства диаграммы в отдельном окне;
4. Marks – метки диаграммы;
5. Legend – легенда диаграммы;
6. Copy – копировать диаграмму в виде рисунка.

Свойства

Свойство	Описание
public Chart Chart	Объект Chart для которого действует эта панель инструментов
public override List<MenuItem> ToolItems	Список элементов панели инструментов
public string DialogsDefaultPath	

Настройка ChartToolbar

При настройке визуального компонента необходимо задать свойство Chart – это диаграмма к которой будет привязана панель инструментов.

```
ChartToolbar chartToolbar = new ChartToolbar();  
chartToolbar.Dock = DockStyle.Top;  
chartToolbar.Parent = this;  
chartToolbar.Chart = chart;
```


DataSource

Компонент **DataSource** – источник данных для куба.

Свойства

Свойство	Описание
public IBaseDataSet DataSet	Набор данных – DBDataSet или DTDataSet
public BaseFields<DataSourceField> Fields	Список полей источника данных

Методы

Метод	Описание
public void AddFields()	Загрузить поля куба из источника данных
public bool Check(StringBuilder msg, bool skipFieldsWithErrors)	Проверить поля на дублирование и другие ошибки
public void Close()	Обнуляет источник данных
public void DeleteFields()	Очищает список полей
public void InitFields(bool loaded = false)	Инициализация полей
public bool Open()	Открыть источник данных. При этом данные будут загружены.

В настройках этого компонента необходимо определить набор данных. Это может быть DBDataSet или DTDataSet.

Ниже, для перечисленных компонентов будет показана настройка из кода приложения всей цепочки получения данных.

Настройка полей DataSource

Настройка полей источника данных нужна, когда требуется конвертировать поле в другой тип данных, форматировать поле, или просто требуется загрузка определенных полей.

Свойство DataSource.Fields хранит описание полей источника. Само поле описывается объектом DataSourceField.

Свойства источника DataFieldProperties зависят от его типа AttributeType:

```
public enum AttributeType
{
    None = 0,
    Custom = 1,
    Reference = 2,
    DateTime = 3
}
```

Свойство `DataField` описывает тип данных, имя и название поля в источнике данных, признак конвертирования с указанием конечного типа данных, имя и название поля в кубе.

Примеры:

```
//Загружаем описание полей
dataSource1.AddFields();

//Изменяем название поля в кубе для поля с индексом 2
dataSource1.Fields[2].DataField.CubeFieldCaption = "Customer";

//Устанавливаем правило конвертирования для поля с именем Population
dataSource1.Fields.GetByName("Population").DataField.Convert = true;
dataSource1.Fields.GetByName("Population").DataField.CubeFieldType = FastReport.Olap.Types.DataType.Int;
```

Настройка атрибутов `DataSource`

Поля данных можно разбивать на атрибуты. Это работает для полей типа дата время и полей зависимых источников.

Для разбивки поля на атрибуты используется свойство поля источника `SplitProperty`. Атрибут может иметь вложенные атрибуты. Уровень вложенности не ограничен.

Для поля можно указывать имя атрибута, содержащего значение `CaptionSource` и имя атрибута для сортировки `OrderSource`.

Пример:

```
FastReport.Olap.Cube.DataSourceField aAttribute;
dataSource1.Fields.GetByName("123").SplitProperty.SplitPaths = FastReport.Olap.Types.DateTimePart.Day;
aAttribute = new
FastReport.Olap.Cube.DataSourceField(dataSource1.Fields.GetByName("Client").SplitProperty.Attributes.Add(
));
```

DBDataSet

DBDataSet – набор данных для DataSource, получаемый из базы данных.

Свойства

Свойство	Описание
<code>public IDbCommand DbCommand</code>	Команда для базы данных, которая содержит SQL запрос

Методы

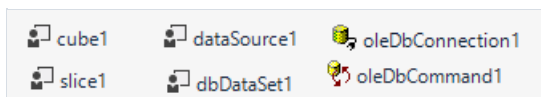
Метод	Описание
<code>public override bool AssignedSource()</code>	Проверка на наличие привязанного источника (DbCommand).
<code>public override void Close()</code>	Закрывает набор данных.
<code>public override bool Open()</code>	Открывает набор данных для чтения.

Настройка подключения куба к базе данных

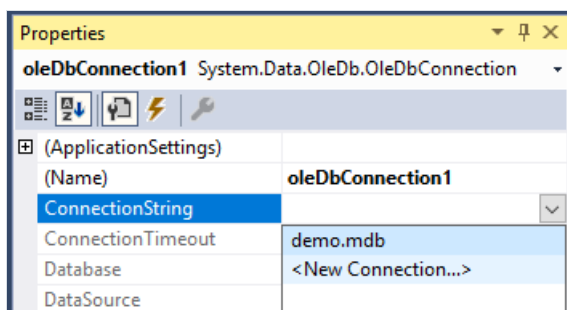
С помощью визуальных компонентов:

Чтобы настроить подключение к данным через DBDataSet требуется сформировать DbCommand с помощью компонента OleDbCommand. В свою очередь, для компонента OleDbCommand требуется задать подключение к базе данных с помощью компонента OleDbConnection.

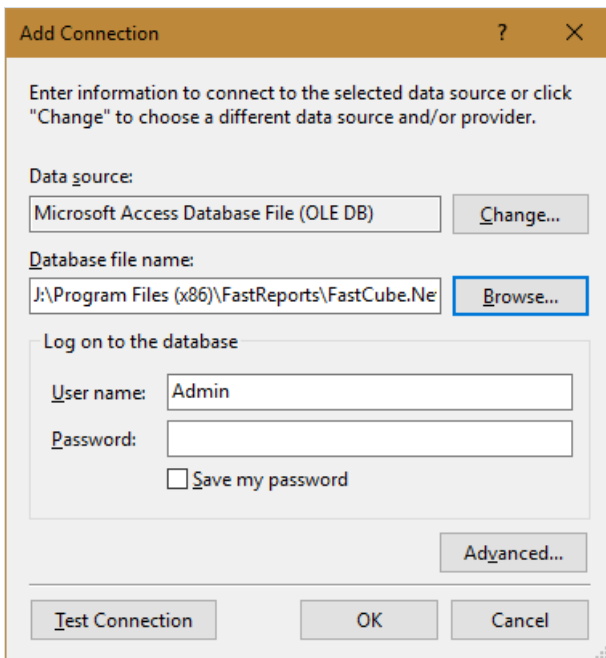
Для настройки подключения к базе данных потребуются следующие компоненты:



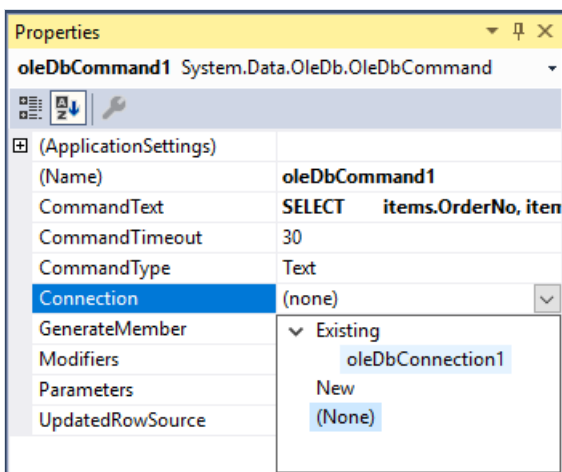
Настройки OleDbConnection:



Создаем новое подключение:

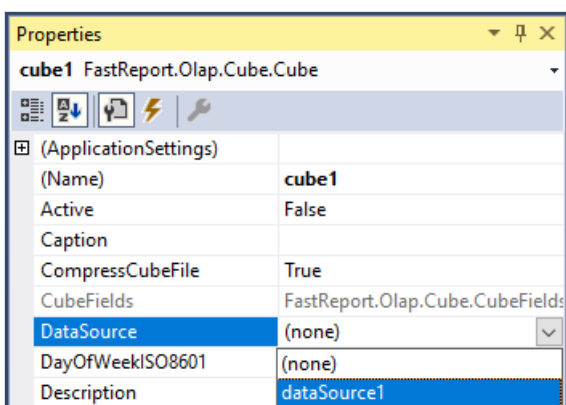


Настройка OleDbCommand:



Необходимо задать команду – sql запрос. А также, нужно выбрать подключение к данным.

Настройка cube:



Из кода приложения

```
OleDbCommand command = new OleDbCommand();  
command.CommandText = "Select * from Sales";  
dbDataSet1.DbCommand = command;  
dataSource1.DataSet = dbDataSet1;  
cube1.DataSource = dataSource1;  
cube1.Open();
```

DTDataSet

DTDataSet – набор данных для DataSource, получаемый из объекта DataTable. В свою очередь DataTable может быть заполнен данными из базы, текстового файла, кода приложения.

Свойства

Свойство	Описание
public DataTable DataTable	Ссылка на таблицу

Методы

Метод	Описание
public override bool AssignedSource()	Проверка на наличие привязанного источника (dataTable).
public override bool Open()	Открыть набор данных для чтения.

Процедура настройки подключения к DataTable с помощью визуальных компонентов очень проста.

Настройка подключения куба к DataTable:

Настройка визуальных компонентов подобна настройке подключения к базе данных, с той лишь разницей, что в настройке DataSet у компонента DataSource нужно выбрать dtDataSet1. Соответственно не требуются OleDbConnection и OleDbCommand.

Из кода приложения:

```
DataTable dataTable = new DataTable(); //Создаем таблицу и наполняем ее данными
...
cube1.Close(); //Закрываем куб, чтобы выгрузить из него данные (если были ранее загружены)
dtDataSet1.DataTable = dataTable; //Для набора данных присваиваем свойству DataTable созданную таблицу
dataSource1.DeleteFields(); //Очищаем поля в источнике данных (если были ранее загружены)
dataSource1.DataSet = dtDataSet1;
cube1.Open(); //Открываем куб, чтобы загрузить в него данные
```

Настройка фильтров

Фильтры служат для отбора данных для расчета в соответствии с заданными критериями.

Управлять фильтрами можно через методы и свойства поля среза.

Примеры:

```
// Начать пакетное изменение фильтра
slice1.SliceFields.GetFieldByIndex(0).BeginUpdateField();

// Закончить пакетное изменение фильтра (применить изменения)
slice1.SliceFields.GetFieldByIndex(0).EndUpdateField();

// Снять признак фильтрации у уникального значения под индексом 3 поля среза с индексом 0
slice1.SliceFields.GetFieldByIndex(0).SetAllowedUniqueValue(3, false);

// Снять признак фильтрации со всех значений поля 'FirstName'
slice1.SliceFields.GetFieldByName("FirstName").SetAllDenied();

// Установить признак фильтрации у значения 'Sergey' поля 'FirstName'
slice1.SliceFields.GetFieldByName("FirstName").SetAllowedUniqueValue(slice1.SliceFields.GetFieldByName("FirstName").CubeField.Values.GetValueIdAtValue("Sergey"), true);

// Установить признак фильтрации у значения с индексом 12 поля 'FirstName'
slice1.SliceFields.GetFieldByName("FirstName").SetAllowedUniqueValue(12, true);

// Устанавливает признак фильтрации только у значения с индексом 4 поля среза с индексом 0
slice1.SliceFields.GetFieldByIndex(0).UniqueValueSingleIndex = 4;

// Инвертирует признак фильтрации значений поля среза с индексом 0
slice1.SliceFields.GetFieldByIndex(0).InverseFilter();

// Устанавливает признак фильтрации поля среза с индексом в соответствии с критерием, заданным в range
slice1.SliceFields.GetFieldByIndex(0).SetRangeFilter(range);

// Установить фильтру тип "переключатель"
slice1.SliceFields.GetFieldByIndex(0).UVFilterType = UniqueValuesFilterType.Single;
```

Управление группами

Группы служат для повышения наглядности представления данных. С их помощью можно группировать значения измерений.

Управлять группами можно через методы и свойства поля среза и менеджера групп (GroupsManager) поля среза.

Примеры:

```
// Создать новую группу MyGroup для поля с индексом 1
var groupIndex =
slice1.SliceFields.GetFieldByIndex(1).GroupsManager.CreateGroup("MyGroup");

// Добавить уникальное значение с индексом 1 в группу индексом groupIndex
slice1.SliceFields.GetFieldByIndex(1).GroupsManager.AddUniqueValueIdToGroup(1, groupIndex);

// Добавить все не сгруппированные значения в группу OtherGroup
slice1.SliceFields.GetFieldByIndex(1).GroupsManager.CreateOtherGroup("OtherGroup");
```


Настройка подсветки показателей

Свойство `MeasureField.Highlights` задает правила подсветки показателей. Правила применяются последовательно, и при одновременной установке одного и того же свойства форматирования двумя и более правилами победит настройка последнего правила из списка.

CustomHighlight

Все правила наследуются от абстрактного класса **CustomHighlight**, который задает лишь базовый каркас.

Свойства

Свойство	Описание
public HighlightApply ApplyTo	Задаёт виды ячеек, к которым можно применять правило.
public HighlightScanDirection ScanDirection	Задаёт направление анализа в котором следует рассматривать соседние ячейки для вычисления при необходимости агрегатов по группе.
public virtual bool IsCustomDrawn	Определяет необходимость отрисовки подсветки классом подсветки. В случае <code>false</code> настройки подсветки задаются через объект <code>Style</code> .
public virtual string Caption	Текстовое представление подсветки для отображения в редакторе.
public virtual bool HideValue	Определяет необходимость скрытия текстовое представление значения.
public CustomHighlights Owner	Ссылка на список <code>Highlights</code> .
public int Index	Индекс правила в списке <code>Highlights</code> .

Методы

Метод	Описание
public virtual void BeginUpdate()	Начать групповую операцию изменения правила. При изменении свойств правила реакция не последует до вызова метода <code>EndUpdate()</code> . Рекомендуется использовать этот метод только если меняется более одного свойства.
public virtual void EndUpdate()	Закончить групповую операцию изменения.
public virtual void Load(XmlItem item)	Загрузить правило из узла XML.
public virtual void Save(XmlItem item)	Сохранить правило в узел XML.

Метод	Описание
public virtual void Assign(CustomHighlight source)	Скопировать настройки из source.
public abstract bool AcceptCell(MeasureCell cell)	Проверка применимости правила к ячейке показателя.
public virtual Expression[] GetExpressions()	Получить массив выражений, требуемый для работы правила (используется внутренними механизмами выполнения выражений).

GraphicHighlight

Абстрактный класс **GraphicHighlight**, от которого фактически наследуются все классы подсветки добавляются следующие свойства и методы.

Свойства

Свойство	Описание
public Style Style	Задаёт стиль ячеек.

Методы

Метод	Описание
public virtual void DrawExample(Graphics g, Rectangle r)	Отрисовать пример подсветки на переданный Graphic в заданную Rectangle область. Используется в редакторе подсветки для отрисовки примера.
public virtual void DrawValue(Graphics g, Rectangle r, MeasureCell Value, ref bool CanDrawImage, ref bool CanDrawText)	Отрисовать значение ячейки. Только для Highlight с IsCustomDrawn = true.
public virtual Style GetStyleFor(MeasureCell value)	Получить стиль отрисовки для ячейки.

Выделение всех ячеек на основании их значений

За функциональность выделения всех ячеек на основании их значений отвечает класс **ContinuousHighlight**. Класс позволяет задать настройки как изменением отдельных свойств, так и применить их из заранее определенного шаблона.

Свойства

Свойство	Описание
public ContinuousHighlightKind Kind	Вид подсветки.
public double MinValue	Минимальное значение (для шкал и гистограммы).

Свойство	Описание
public double MaxValue	Максимальное значение (для шкал и гистограммы).
public double MidValue	Среднее значение (для трехцветной шкалы).
public ContinuousHighlightValueType MinValueType	Тип минимального значения.
public ContinuousHighlightValueType MaxValueType	Тип максимального значения.
public ContinuousHighlightValueType MidValueType	Тип среднего значения.
public Color MinValueColor	Цвет минимального значения.
public Color MaxValueColor	Цвет максимального значения.
public Color MidValueColor	Цвет среднего значения.
public Color BarColor	Цвет гистограммы.
public Color FrameColor	Цвет рамки гистограммы.
public bool GradientDraw	Использовать градиент для гистограммы.
public bool ShowCellValue	Показывать значение ячейки (для гистограммы и набора иконок).
public string IconSet	Текущий набор иконок для типа IconSet.
public ImageList Images	Ссылка на полный набор картинок, из которого загружаются иконки для типа IconSet.
public int ImageCount	Количество иконок для типа IconSet.

Методы

Метод	Описание
public void DrawIconSet(Graphics g, Rectangle r, string name)	Отрисовать набор иконок.
public int GetImageIndex(int icon)	Получить индекс изображения для переданного индекса иконки из текущего набора иконок.
public void SetImageIndex(int icon, int imageIndex)	Установить индекс изображения для переданного индекса иконки из текущего набора иконок.
public double GetImageValue(int icon)	Получить значение для переданного индекса иконки из текущего набора иконок.

Метод	Описание
public void SetImageValue(int icon, double value)	Установить значение для переданного индекса иконки из текущего набора иконок.
public ContinuousHighlightValueType GetImageValueType(int icon)	Получить тип значения для переданного индекса иконки из текущего набора иконок.
public void SetImageValueType(int icon, ContinuousHighlightValueType value)	Установить тип значения для переданного индекса иконки из текущего набора иконок.
public ContinuousHighlightIconCondition GetImageValueCondition(int icon)	Получить тип сравнения для переданного индекса иконки из текущего набора иконок.
public void SetImageValueCondition(int icon, ContinuousHighlightIconCondition value)	Установить тип сравнения для переданного индекса иконки из текущего набора иконок.
public string GetImageInfo(int icon)	Получить текстовое представление условия для переданного индекса иконки для отображения в редакторе.
public void SetImageReverseOrder()	Поменять порядок следования иконок для текущего набора иконок.
public bool LoadPreset(ContinuousHighlightKind kind, string name)	Загрузить настройки подсветки из шаблона.

Определены следующие шаблоны подсветки (в зависимости от типа).

ContinuousHighlightKind.TwoColorScale:

- White - Red
- Red - White
- Green - White
- White - Green
- Green - Yellow
- Yellow - Green

ContinuousHighlightKind.ThreeColorScale:

- Green - Yellow - Red
- Red - Yellow - Green
- Green - White - Red
- Red - White - Green
- Blue - White - Red
- Red - White - Blue

ContinuousHighlightKind.BarChart:

- Blue

- Green
- Red
- Orange
- Light Blue
- Purple
- Blue Gradient
- Green Gradient
- Red Gradient
- Orange Gradient
- Light Blue Gradient
- Purple Gradient

ContinuousHighlightKind.IconSet:

- Arrows (Colored)
- Arrows (Gray)
- Flags (Colored)
- 3 Traffic Lights (Unrimmed)
- 3 Traffic Lights (Rimmed)
- 3 Signs
- 3 Symbols (Circled)
- 3 Symbols (Uncircled)
- 3 Stars
- 3 Triangles
- Arrows (Colored)
- 4 Arrows (Gray)
- Red To Black
- 4 Ratings
- 4 Traffic Lights
- Arrows (Colored)
- 5 Arrows (Gray)
- 5 Ratings
- 5 Quarters
- 5 Boxes

Пример:

```
// Создаем новую подсветку ContinuousHighlight для первого показателя
ContinuousHighlight h = new ContinuousHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Загружаем из шаблона BarChart "Orange Gradient"
h.LoadPreset(ContinuousHighlightKind.BarChart, "Orange Gradient");

// Создаем новую подсветку ContinuousHighlight для второго показателя
ContinuousHighlight h = new ContinuousHighlight(slice1.MeasuresContainer.MeasureFields[1].Highlights);

// Загружаем из шаблона IconSet "3 Flags (Colored)"
h.LoadPreset(ContinuousHighlightKind.IconSet, "3 Flags (Colored)");
```

Выделение только ячеек попадающих под условие

RangeHighlight - отбор ячеек, по условию в свойстве Range.

Свойства

Свойство	Описание
<code>public Range Range</code>	Условие отбора.

Пример

```
// Создаем новую подсветку RangeHighlight с подсветкой ячеек Fuchsia для значений > 1000
RangeHighlight h = new RangeHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Начинаем группу изменений
h.BeginUpdate();

// Задаем цвет заливки ячеек
h.Style.FillColor = Color.Fuchsia;

// Сравниваем по значению ячейки
h.Range.CompareObject = RangeCompareObject.Value;

// Условие >
h.Range.ValueCondition = RangeValueCondition.Greater;

// Сравниваем со значением 1000
h.Range.LowRange = 1000;

// Заканчиваем группу изменений
h.EndUpdate();
```

Выделение первых или последних ячеек

TopHighlight - отбор первых или последних ячеек.

Свойство	Описание
<code>public TopType TopType</code>	Первые или последние ячейки.

Свойство	Описание
public int TopCount	Количество первых или последних.
public bool TopPercent	Признак использования процентов.

Пример

```
// Создаем новую подсветку TopHighlight подсветкой Fuchsia для первых 30% значений
TopHighlight h = new TopHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Начинаем группу изменений
h.BeginUpdate();

// Задаем цвет заливки ячеек
h.Style.FillColor = Color.Fuchsia;

// Задаем первые ячейки
h.TopType = TopType.Мах;

// Задаем количество
h.TopCount = 30;

// Указываем что количество в процентах
h.TopPercent = true;

// Заканчиваем группу изменений
h.EndUpdate();
```

Выделение ячеек относительно среднего

AverageHighlight - отбор значений с отклонением от среднего.

Свойство	Описание
public AverageValueCondition Condition	Условие сравнения со средним значением.
public decimal StdDev	Количество стандартных отклонений от среднего.

Пример

```

// Создаем новую подсветку AverageHighlight подсветкой Fuchsia для значений больших среднего на 0.1
стандартных отклонения

AverageHighlight h = new AverageHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Начинаем группу изменений
h.BeginUpdate();

// Задаем цвет заливки ячеек
h.Style.FillColor = Color.Fuchsia;

// Задаем условие сравнения
h.Condition = AverageValueCondition.Above;

// Задаем количество стандартных отклонений
h.StdDev = 0.1M;

// Заканчиваем группу изменений
h.EndUpdate();

```

Выделение повторяющихся или уникальных значений

UniqueHighlight - отбор повторяющихся и уникальных значений.

Свойство

Описание

Свойство	Описание
public UniqueValueCondition Condition	Условие сравнения.

Пример

```

// Создаем новую подсветку UniqueHighlight подсветкой Fuchsia для повторяющихся значений
UniqueHighlight h = new UniqueHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Начинаем группу изменений
h.BeginUpdate();

// Задаем цвет заливки ячеек
h.Style.FillColor = Color.Fuchsia;

// Задаем условие сравнения
h.Condition = UniqueValueCondition.Repeatable;

// Заканчиваем группу изменений
h.EndUpdate();

```

Выделение ячеек, попадающих под выражение

ExpressionHighlight - отбор значений по условию Expression.

Свойство

Описание

Свойство	Описание
public string Expression	Выражение сравнения.

Пример


```
// Создаем новую подсветку ExpressionHighlight с подсветкой Fuchsia для выражения
ExpressionHighlight h = new ExpressionHighlight(slice1.MeasuresContainer.MeasureFields[0].Highlights);

// Начинаем группу изменений
h.BeginUpdate();

// Задаем цвет заливки ячеек
h.Style.FillColor = Color.Fuchsia;

// Задаем условие сравнения
h.Expression = "Substring([Manager], 0, 1)= \"B\"";

// Заканчиваем группу изменений
h.EndUpdate();
```

Информация о выпусках

[Версия 2020.2](#)

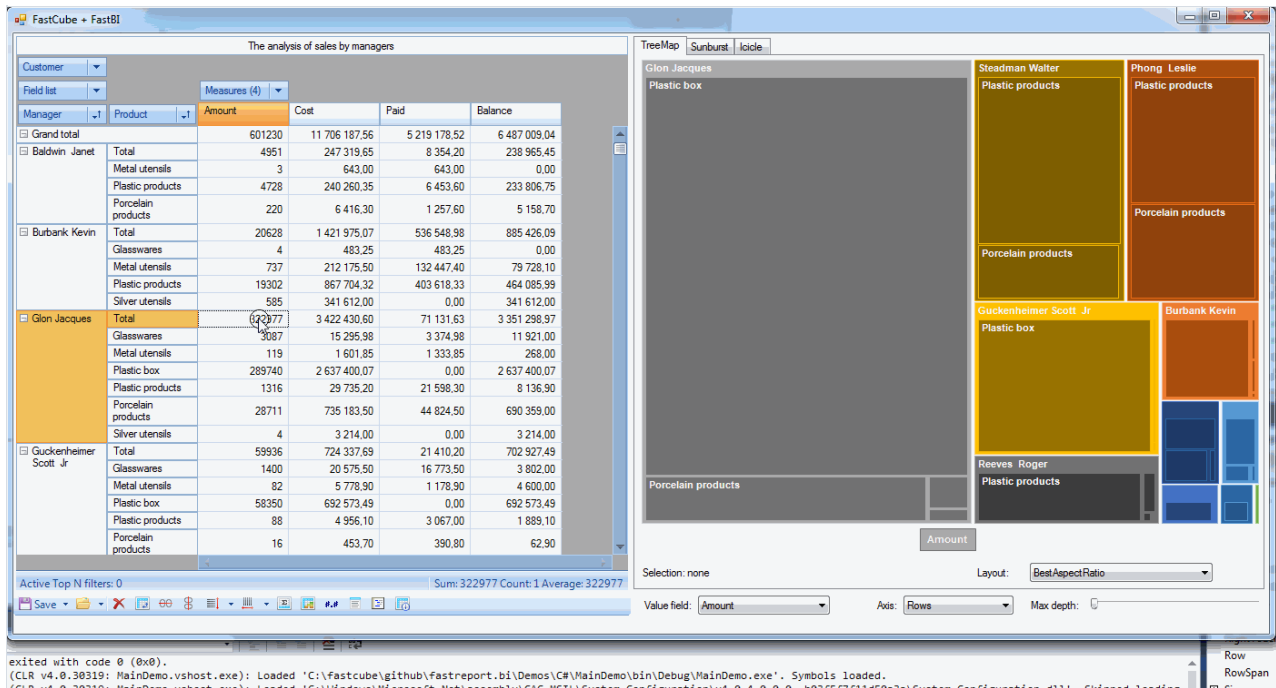
[Версия 2020.1](#)

[Версия 2019.4](#)

Информация о выпуске FastCube Net 2021.1

Новые возможности

Добавлено расширение FastReport.BG



Добавлен компонент ListDataSet позволяющий загружать в куб данные используя Windows Forms Data Binding

Добавлена возможность загружать данные в куб используя механизм Windows Forms Data Binding, использующий стандартные элементы управления: DataGridView, ListView и Combobox.

Windows Forms Data Binding работает с данными посредством компонента `BindingSource`, который выступает в роли прокси (больше информации по теме можно найти в MSDN).

Далее приведен небольшой пример загрузки данных из `List<>`.

1. Объявляем класс ProductInfo

```
public class ProductInfo
{
    public String Name { get; set; }
    public String Group { get; set; }
    public int Count { get; set; }
}
```

2. Создаем список записей ProductInfo

```

List<ProductInfo> CreateProducts()
{
    return new List<ProductInfo> {
        new ProductInfo {Group = "Bakery products", Name = "Ciabatta", Count = 3},
        new ProductInfo {Group = "Bakery products", Name = "Bread", Count = 5},
        new ProductInfo {Group = "Bakery products", Name = "Croissant", Count = 1},

        new ProductInfo {Group = "Alcohol", Name = "Wine", Count = 6},
        new ProductInfo {Group = "Alcohol", Name = "Whiskey", Count = 2},
        new ProductInfo {Group = "Alcohol", Name = "Beer", Count = 5},

        new ProductInfo {Group = "Dairy products", Name = "Yoghurt", Count = 5},
        new ProductInfo {Group = "Dairy products", Name = "Milk", Count = 4}
    };
}

```

3. Настраиваем Cube и slice

```

private void Form1_Load(object sender, EventArgs e)
{
    // настраиваем источник данных
    listDataSet1.DataSource = CreateProducts();
    // подгружаем данные в Cube
    cube1.Active = true;
    // настраиваем Slice
    slice1.YAxisContainer.AddSliceField(slice1.SliceFields.GetFieldByName("Group"));
    slice1.YAxisContainer.AddSliceField(slice1.SliceFields.GetFieldByName("Name"));
    var measure = new FastReport.Olap.Slice.MeasureField(slice1,
FastReport.Olap.Types.AggregateFunction.Sum, slice1.SliceFields.GetFieldByName("Count"));
    slice1.MeasuresContainer.AddMeasure(measure);
    slice1.XAxisContainer.AddMeasuresField();
}

```

В результате получаем следующий вид:

Move filter fields here		
Field list		Measures (1)
Group	Name	Count
<input type="checkbox"/> Grand total		31
<input type="checkbox"/> Alcohol	Total	13
	Beer	5
	Whiskey	2
	Wine	6
<input type="checkbox"/> Bakery products	Total	9
	Bread	5
	Ciabatta	3
	Croissant	1
<input type="checkbox"/> Dairy products	Total	9
	Milk	4
	Yoghurt	5

Полный пример расположен в каталоге `Demos\C#\DataBindings`.

Другие изменения

- В контекстное меню XAxisZone, YAxisZone компонента SliceGrid добавлен пункт Copy, копирующий в буфер обмена заголовок выбранного узла
- В выпадающих списках вставка из буфера осуществляет поиск значения
- При двойном клике по значению в выпадающем фильтре измерения осуществляется

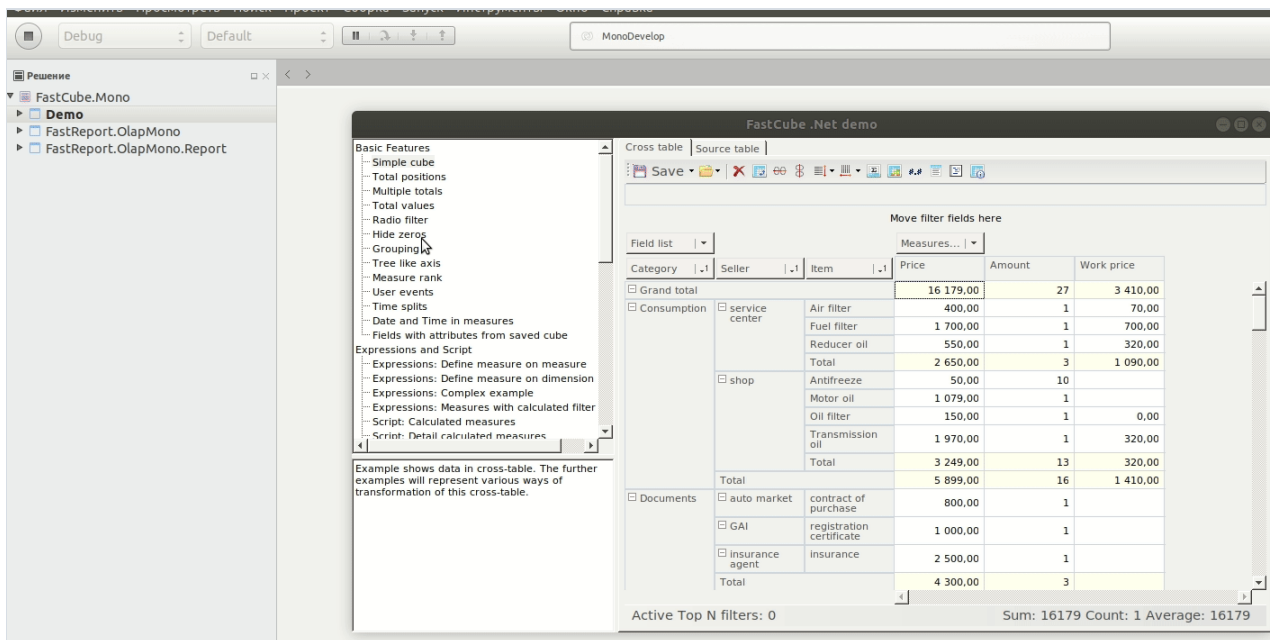
позиционирование на это значение в SliceGrid

Информация о выпуске FastCube .NET 2020.2

Новые возможности

Добавлена поддержка платформы Mono

Начиная с этой версии мы добавляем поддержку Mono.



Значительные изменения

- До этого релиза мы поставляли пакет FastCube.Olap с зависимостью от библиотеки FastReport. Сейчас мы вынесли эту зависимость и собственно интеграцию с FastReport в отдельный пакет FastReport.Olap.Report для платформы .Net и FastReport.Olap.ReportMono для платформы Mono.

Другие изменения

- Редактор выражений теперь показывает имена измерений/показателей/полей вместо их заголовков.

Ошибки

- Ошибки в выражениях не вызывают Exception
- Ошибки в выражении ExpressionHighlight не вызывают Exception
- Ошибка добавления элемента в группу
- Ошибка отображения значений на диаграммах в некоторых локалях
- Поправлен расчет агрегата "List of values"
- Поправлен расчет агрегата "Median"

Информация о выпуске FastCube .NET 2020.1

Новые возможности

Добавлена поддержка ASP.Net Core

Начиная с этой версии мы добавляем поддержку ASP.Net Core. К использованию подготовлены 2 интерактивных Web-контроля: WebSliceGrid и WebCubeGrid - Web-аналоги SliceGrid и CubeGrid.

FastCube.Web for Core

About FastCube.Net Download Demo Buy FastCube.Net

Simple cube

Total positions
Grouping
Calculated measures
Query from DB table

Drop fields here				Measures (3)		
Field list	Category	Seller	Item	Price	Amount	Work price
Grand total				16,179.00	27	3,410.00
Consumption	service center		Air filter	400.00	1	70.00
			Fuel filter	1,700.00	1	700.00
			Reducer oil	550.00	1	320.00
			Total	2,650.00	3	1,090.00
	shop		Antifreeze	50.00	10	
			Motor oil	1,079.00	1	
			Oil filter	150.00	1	0.00
			Transmission oil	1,970.00	1	320.00
			Total	3,249.00	13	320.00
	Total			5,899.00	16	1,410.00
Documents	auto market		contract of	800.00	1	
	GAI		registration	1,000.00	1	
	insurance agent		insurance	2,500.00	1	
	Total			4,300.00	3	
Goods	prev owner		Winter cover	1,100.00	4	500.00
	shop		car kit	1,000.00	1	
			car number	80.00	2	
			signalling	3,800.00	1	1,500.00
			Total	4,880.00	4	1,500.00
	Total			5,980.00	8	2,000.00

© 2020 - Fast Reports, Inc.

Онлайн-демо: <https://www.fast-report.com:2020/>

Подсветка ячеек показателей

В этой версии мы значительно расширили возможности FastCube .Net по подсветке ячеек. Мы добавили новые классы подсветок, расширили возможности ранее существовавших классов и исправили ряд ошибок.

- Класс TopHighlight - выделение наибольших и наименьших значений ячеек
- Класс AverageHighlight - выделение ячеек относительно средних значений
- Класс UniqueHighlight - выделение повторяющихся и уникальных значений ячеек
- Класс ExpressionHighlight - выделение ячеек на основе выражений

The analysis of sales by managers

Customer	Field list	Measures (5)	Amount	Cost	Paid	Balance	All paid
Grand total			601230	11 706 187,56	5 219 178,52	6 487 009,04	False
Baldwin Janet	Total		4951	247 319,65	8 354,20	238 965,45	False
	Metal utensils		3	643,00	643,00	0,00	True
	Plastic products		4				
	Porcelain products						
Burbank Kevin	Total		20				
	Glasswares						
	Metal utensils						
	Plastic products		19				
	Silver utensils						
	Total		322				
Glon Jacques	Glasswares		3				
	Metal utensils						
	Plastic box		289				
	Plastic products		1316	29 735,20	21 598,30	8 136,90	False
	Porcelain products		28711	735 183,50	44 824,50	690 359,00	False
	Silver utensils		4	3 214,00	0,00	3 214,00	False
	Total		59936	724 337,69	21 410,20	702 927,49	False
	Glasswares		1400	20 575,50	16 773,50	3 802,00	False
	Metal utensils		82	5 778,90	1 178,90	4 600,00	False
	Plastic box		58350	692 573,49	0,00	692 573,49	False
Guckenheimer Scott Jr	Plastic products		88	4 956,10	3 067,00	1 889,10	False
	Porcelain products		16	453,70	390,80	62,90	False
	Total		6957	364 227,36	1 592,20	362 635,16	False
	Metal utensils		401	12 929,70	0,00	12 929,70	False
Montgomery Jennifer	Plastic products		5134	297 966,46	1 461,60	296 504,86	False

- Мы также дополнили возможности гистограммы (класс ContinuousHighlight) представлением отрицательных значений: отрицательные и положительные значения могут быть отрисованы разным стилем и по разную сторону оси.

The analysis of sales by managers

Customer	Field list	Measures (5)	Amount	Cost	Paid	Balance	Test
Grand total			601230	11 706 187,56	5 219 178,52	6 487 009,04	601230
Baldwin Janet	Total		4951	247 319,65	8 354,20	238 965,45	-4951
	Metal utensils		3	643,00	643,00	0,00	-3
	Plastic products		4728	240 260,35	6 453,60	233 806,75	4728
	Porcelain products		220	6 416,30	1 257,60	5 158,70	220
Burbank Kevin	Total		20628	1 421 975,07	536 548,98	885 426,09	20628
	Glasswares		4	483,25	483,25	0,00	4
	Metal utensils		737	212 175,50	132 447,40	79 728,10	-737
	Plastic products		19302	867 704,32	403 618,33	464 085,99	19302
	Silver utensils		585	341 612,00	0,00	341 612,00	-585
Glou Jacques	Total		322977	3 422 430,60	71 131,63	3 351 298,97	-322977
	Glasswares		3087	15 295,98	3 374,98	11 921,00	-3087
	Metal utensils		119	1 601,85	1 333,85	268,00	-119
	Plastic box		289740	2 637 400,07	0,00	2 637 400,07	289740
	Plastic products		1316	29 735,20	21 598,30	8 136,90	1316
	Porcelain products		28711	735 183,50	44 824,50	690 359,00	-28711
	Silver utensils		4	3 214,00	0,00	3 214,00	4
Guckenheimer Scott Jr	Total		59936	724 337,69	21 410,20	702 927,49	59936
	Glasswares		1400	20 575,50	16 773,50	3 802,00	1400
	Metal utensils		82	5 778,90	1 178,90	4 600,00	82
	Plastic box		58350	692 573,49	0,00	692 573,49	58350
	Plastic products		88	4 956,10	3 067,00	1 889,10	88
	Porcelain products		16	453,70	390,80	62,90	16
Montgomery	Total		6957	364 227,36	1 592,20	362 635,16	-6957

Active Top N filters: 0 | Sum: 220 Count: 1 Average: 220

Пожалуйста, обратитесь к нашей обновленной документации по FastCube .Net и посмотрите детальное описание и примеры использования.

Сохранение во внешние сервисы (облака)

Мы добавили возможности сохранения куба, среза и экспортов в Google Drive, DropBox и FTP

Basic Features

- Simple cube
- Total positions
- Multiple totals
- Total values
- Radio filter
- Hide zeros
- Grouping
- Tree like axis
- Measure rank
- User events
- Time splits
- Date and Time in measures
- Fields with attributes from saved cube

Expressions and Script

- Expressions: Define measure on measure
- Expressions: Define measure on dimension
- Expressions: Complex example
- Expressions: Measures with calculated filter
- Script: Calculated measures
- Script: Detail calculated measures
- Script: Measures with calculated filter

Data Highlight

- Continous Highlight
- Range Highlight

Example shows data in cross-table. The further examples will represent various ways of transformation of this cross-table.

Cross table Source table Chart

Save

Move filter fields here

Field list	Seller	Item	Price	Amount	Work price
Grand total			16 179,00	27	3 410,00
Consumption	service center	Air filter	400,00	1	70,00
		Fuel filter	1 700,00	1	700,00
		Reducer oil	550,00	1	320,00
		Total	2 650,00	3	1 090,00
	shop	Antifreeze	50,00	10	
		Motor oil	1 079,00	1	
		Oil filter	150,00	1	0,00
		Transmission oil	1 970,00	1	320,00
		Total	3 249,00	13	320,00
	Total		5 899,00	16	1 410,00
Documents	auto market	contract of purchase	800,00	1	
	GAI	registration certificate	1 000,00	1	
	insurance agent	insurance	2 500,00	1	
	Total		4 300,00	3	
Goods	prev owner	Winter cover	1 100,00	4	500,00
	shop	car kit	1 000,00	1	
		car number adapter	80,00	2	
		signalling	3 800,00	1	1 500,00
		Total	4 880,00	4	1 500,00

Active Top N filters: 0 | Sum: 16179 Count: 1 Average: 16179

Значительные изменения

- Меню экспорт удалено, а варианты экспорта реорганизованы и перенесены в меню "Сохранить"
- Перенос классов Highlight в пространство имен FastReport.Olap.Highlight
- Значительный рефакторинг ContinuousHighlight с потерей совместимости: Ранее: ContinuousHighlightValueType содержал элементы BoundByRow, BoundByCol, PercentByRow, PercentByCol, PercentileByRow, PercentileByCol. Теперь: элементы заменены на Bound, Percent, Percentile. Дополнительно введен тип HighlightScanDirection и соответствующее свойство ScanDirection общее для всех типов Highlight со значениями Auto, Rows, Columns. Новые сохранения не будут корректно работать в старой версии и в текущей версии FastCube VCL 2. Эффект: упрощение настройки подсветки в связи с уменьшением числа вариантов выпадающих списков, значение Auto подходит для большинства случаев и сохраняет работоспособность Highlight при переносе поля "Показатели" из столбцов в строки и наоборот.

Ошибки

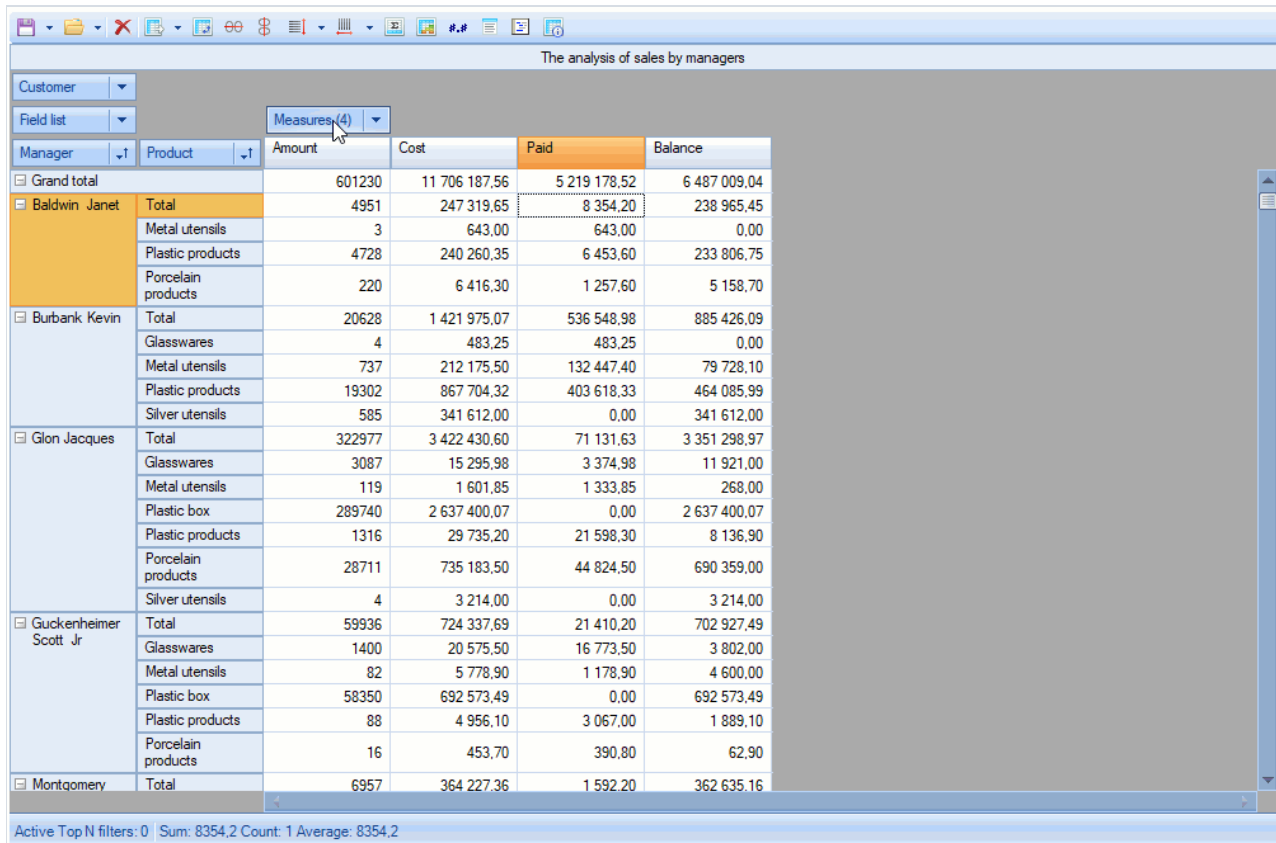
- Исправлена ошибка копирования в буфер ячеек из области данных SliceGrid.
- Исправлена ошибка построения диаграммы.

Информация о выпуске FastCube .NET 2019.4

Новые возможности

Выражения

С этим релизом в FastCube Net добавлены выражения. До сих пор вычисляемые показатели или итоги можно было создать с помощью скрипта на C# или VB.Net. Это не было не удобно и не понятно пользователям далеким от программирования. Таким образом, мы решили добавить в продукт поддержку выражений, которые уже могут быть вам знакомы по нашему продукту FastReport.



Customer	Manager	Product	Amount	Cost	Paid	Balance
Grand total			601230	11 706 187,56	5 219 178,52	6 487 009,04
Baldwin Janet		Total	4951	247 319,65	8 354,20	238 965,45
		Metal utensils	3	643,00	643,00	0,00
		Plastic products	4728	240 260,35	6 453,60	233 806,75
		Porcelain products	220	6 416,30	1 257,60	5 158,70
Burbank Kevin		Total	20628	1 421 975,07	536 548,98	885 426,09
		Glasswares	4	483,25	483,25	0,00
		Metal utensils	737	212 175,50	132 447,40	79 728,10
		Plastic products	19302	867 704,32	403 618,33	464 085,99
		Silver utensils	585	341 612,00	0,00	341 612,00
Glon Jacques		Total	322977	3 422 430,60	71 131,63	3 351 298,97
		Glasswares	3087	15 295,98	3 374,98	11 921,00
		Metal utensils	119	1 601,85	1 333,85	268,00
		Plastic box	289740	2 637 400,07	0,00	2 637 400,07
		Plastic products	1316	29 735,20	21 598,30	8 136,90
		Porcelain products	28711	735 183,50	44 824,50	690 359,00
		Silver utensils	4	3 214,00	0,00	3 214,00
Guckenheimer Scott Jr		Total	59936	724 337,69	21 410,20	702 927,49
		Glasswares	1400	20 575,50	16 773,50	3 802,00
		Metal utensils	82	5 778,90	1 178,90	4 600,00
		Plastic box	58350	692 573,49	0,00	692 573,49
		Plastic products	88	4 956,10	3 067,00	1 889,10
		Porcelain products	16	453,70	390,80	62,90
Montgomery		Total	6957	364 227,36	1 592,20	362 635,16

Active Top N filters: 0 Sum: 8354,2 Count: 1 Average: 8354,2

Пожалуйста, обратитесь к нашей обновленной документации по FastCube .Net и посмотрите новые примеры для лучшего понимания что такое выражения и как они работают.

Диалог "свойства куба"

Мы заменили диалоговое окно **Информация** на новое диалоговое окно **свойства куба**, в которое добавили возможность переключить язык скрипта и настраивать список assemblies для куба.

Category	Seller	Item	Price	Amount	Work price
Grand total			16 179.00	27	3 410.00
Consumption	service center	Air filter	400.00	1	70.00
		Fuel filter	1 700.00	1	700.00
		Reducer oil	550.00	1	320.00
		Total	2 650.00	3	1 090.00
	shop	Antifreeze	50.00	10	
		Motor oil	1 079.00	1	
		Oil filter	150.00	1	0.00
		Transmission oil	1 970.00	1	320.00
		Total	3 249.00	13	320.00
		Total	5 899.00		16
Documents	auto market	contract of purchase	800.00	1	
	GAI	registration certificate	1 000.00	1	
	insurance agent	insurance	2 500.00	1	

Active Top N filters: 0 | Sum: 16179 Count: 1 Average: 16179

Формат вывода для измерений

Мы добавили возможность установить формат вывода для измерений. Это может пригодиться для не строковых измерений. Используйте свойство `AxisField.DisplayFormat` или редактор измерения чтобы настраивать формат вывода.

Date	Amount
Grand total	30
January 05	78
January 10	90
January 11	3
January 12	60
January 13	31
January 14	30
January 17	01
January 18	19
January 19	25
January 20	14
January 21	23
January 24	75
January 25	70
January 26	2
January 27	51
January 30	78
January 31	10
February 01, 2006	720
February 02, 2006	1614
February 03, 2006	448
February 06, 2006	3
February 07, 2006	659
February 08, 2006	33020
February 09, 2006	1164

Привязка заголовка измерения и содержимого

С появлением свойств `AxisContainer.BindCaptionWidth` и `Slice.BindCaptionWidth` стало возможно привязать ширину заголовка измерения и его содержимого в оси.

BindCaptionWidth = False

Category	Seller	Item	Price
Grand total			16 179,00
Consumption	service center	Air filter	400,00
		Fuel filter	1 700,00
		Reducer oil	550,00
		Total	2 650,00
	shop	Antifreeze	50,00
		Motor oil	1 079,00
		Oil filter	150,00
		Transmission oil	1 970,00
		Total	3 249,00

BindCaptionWidth = True

Category	Seller	Item	Price
Grand total			16 179,00
Consumption	service center	Air filter	400,00
		Fuel filter	1 700,00
		Reducer oil	550,00
		Total	2 650,00
	shop	Antifreeze	50,00
		Motor oil	1 079,00
		Oil filter	150,00
		Transmission oil	1 970,00
		Total	3 249,00

Опция IgnoreCase для выражений диапазонов

Мы используем выражения диапазонов для фильтрации измерений для правил подсветки показателей. Начиная с этого релиза стало возможно игнорировать регистр для текстовых сравнений в выражениях для диапазонов.

Свойство `Range.IgnoreCase` можно поменять как из программного кода, так и в редакторе диапазонов.

Customer	Manager	Product	Amount	Cost	Paid	Balance
Grand total		Total	601230	11 706 187,56	5 219 178,52	6 487 009,04
Baldwin Janet		Total	4951	247 319,65	8 354,20	238 965,45
		Metal utensils	3	643,00	643,00	0,00
		Plastic products	4728	240 260,35	6 453,60	233 806,75
		Porcelain products	220	6 416,30	1 257,60	5 158,70
Burbank Kevin		Total	20628	1 421 975,07	536 548,98	885 426,09
		Glasswares	4	483,25	483,25	0,00
		Metal utensils	737	212 175,50	132 447,40	79 728,10
		Plastic products	19302	867 704,32	403 618,33	464 085,99
		Silver utensils	585	341 612,00	0,00	341 612,00
Glon Jacques		Total	322977	3 422 430,60	71 131,63	3 351 298,97
		Glassware	3087	15 295,98	3 374,98	11 921,00
		Metal utensils	119	1 601,85	1 333,85	268,00
		Plastic box	289740	2 637 400,07	0,00	2 637 400,07
		Plastic products	1316	29 735,20	21 598,30	8 136,90
		Porcelain	8874	705 400,50	11 881,50	888 282,00

Active Top N filters: 0 | Sum: 867704,32 Count: 1 Average: 867704,32

Другие измерения

- Добавлены методы `PopupWindow.ExpandAll()` и `PopupWindow.CollapseAll()`, а также свойство `CustomGrid.AutoExpandPopups` (по умолчанию = true), которое определяет показывать ли выпадающие списки грида автоматически свернутыми или развернутыми.
- Добавлена возможность установки собственных контекстных меню (`ContextMenuStrip`) для зон грида.
- Добавлены кнопки очистки в редактор показателя.
- Добавлен пункт в контекстное меню оси для удаления показателя (при нажатии на заголовок показателя).

Ошибки

- Поправлены ошибки с доступом к значению измерений из скрипта.

Контакты и техподдержка

Вы всегда можете задать вопросы по использованию продукта с помощью [email](#), либо с помощью [формы на сайте](#).

Также мы будем рады вашим предложениям по улучшению нашего продукта.