

# Мини-учебник по ReoGrid версии 2.1

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Наименование</b>	<b>Цена за ед.</b>	<b>Кол-во</b>	<b>Итоговая цена</b>							
2	<i>Atmel AT89S52-24PC</i>	5	4	20							
3	<i>Raspberry Pi 2 Model B</i>	38	3	114							
4	<i>UNO R3 KIT</i>	25	2	50							

## Оглавление

### Первое знакомство

- Поддерживаемые платформы, форматы файлов, издания и редакции
- Подключение или обновление ReoGrid

### Введение

#### Пора начинать

- Пространства имён
- Книга в памяти (виртуальная книга)
- Манипулирование листами книги
- Основные (простые) действия над ячейками и диапазонами ячеек
- Типы ячеек
- Форматы данных
- Встроенные шаблоны форматирования данных
- Управление параметрами книги или листа
- Обработка событий
- Внешний вид компонента и локализация
- Инициализация списков
- Совместимость с форматом Excel
- Формула (Formula)

#### Раздел "Файл"

- Сброс таблицы или листа
- Загрузка данных
- Сохранение данных
- Печать

#### Раздел "Правка"

- Отмена последнего действия, возврат отменённого действия и повтор последнего
- Буфер обмена и "Выделить все"

#### Раздел "Вид"

- Фиксация ячеек
- Изменение размеров таблицы
- **Стили**
- Шрифты и содержимое ячеек
- Перенос слов
- Выравнивание
- Заливка фона
- Поворот текста в ячейке
- Границы (Borders)
- Получение данных о выбранном диапазоне
- Объединение, разбивка и группировка ячеек
- Масштабирование

#### Диаграммы

- Линейная диаграмма
- Вертикальная гистограмма
- Горизонтальная гистограмма
- Круговая 2D диаграмма
- Диаграмма с областями
- Кольцевая диаграмма

#### Плавающие объекты

Отображение текста в формате **RTF**

Автозаполнение

## Дополнительные примеры

- Настраиваемые заголовки
- Определение разрешённого для редактирования диапазона
- Использование книги в памяти
- Создание редактора электронных таблиц начального уровня

# Первое знакомство

## Поддерживаемые платформы

**ReoGrid** поддерживает технологии **Windows Form** и **WPF**. Поэтому в архиве с файлами компонента приведено два варианта компонента: WPF-версия содержит эти три буквы в названии папки.

Все редакции **ReoGrid** поддерживают клиентский профиль (**.NET Client profile**) версии **3.5**. Код примеров данного руководства выполнялся в **.NET Client profile 4**.

## Поддерживаемые форматы файлов

ReoGrid в разной степени поддерживает 4 формата файлов.

Формат	Пояснение
<b>rgf</b>	Собственный формат хранения данных, в его основе лежит расширяемый язык разметки XML. Данные можно как сохранять, так и загружать.
<b>xlsx</b>	Формат хранения данных Microsoft Excel 2007 и более новых версий. Данные можно как сохранять, так и загружать.
<b>csv</b>	Простой формат для хранения табличных данных в текстовом файле. Данные можно как сохранять, так и загружать, но только для одного листа.
<b>html</b>	Документ на языке гипертекстовой разметки. Данные листа можно лишь экспортировать в данный формат, загрузка данных из такого файла не предусмотрена.

## Особенности распространения и поддержки

ReoGrid, начиная с версии 1.3.0, распространяется под лицензией MIT. Из платных возможностей остались: техническая поддержка на русском, украинском, английском, японском и китайском языках и запросы на функционал под заказ. Подробнее можно узнать этой на странице (<http://reogrid.net/ru/purchase/>).

## Выбор редакции

Компонент ReoGrid доступен в нескольких редакциях

Редакция	Пояснение
<b>Минимальная (Minimum или WPFMinimum)</b>	Содержит лишь библиотеку <b>unvell.ReoGrid.dll</b> . Доступны все основные функции.
<b>Стандартная (Release)</b>	Все основные функции и вычисление формул
<b>Расширенная (Extension или WPFExtension)</b>	Помимо указанных выше, содержит библиотеки для работы со сценариями: <b>unvell.ReoScript.dll</b> и <b>unvell.ReoScript.EditorLib.dll</b> . Также в состав этой редакции входят два сторонних компонента: <b>Antlr3.Runtime.dll</b> и <b>FastColoredTextBox.dll</b> (Windows Form).

## Подключение или обновление ReoGrid

### Первое подключение

1. Копируем библиотеки компонента в папку **Debug** и/или **Release** вашего проекта.
2. В свойствах проекта, разделе "**Ссылки**" подключаем библиотеки к проекту.
3. В свойствах панели элементов управления выбираем пункт "Выбрать компоненты", указываем на библиотеку **ReoGrid** и выбираем подключаемые компоненты.
4. Если всё сделано верно, то в панели элементов управления появится **ReoGrid**.

### Обновление

1. Переходим к свойствам проекта. В разделе "Ссылки" удаляем все ссылки на библиотеки ReoGrid.
2. Из папки проекта **Debug** и/или **Release** вашего проекта удаляем файлы устаревшей сборки ReoGrid.
3. Копируем файлы новой версии в папку **Debug / Release** вашего проекта.
4. В свойствах проекта, разделе "Ссылки" добавляем ссылки на библиотеки ReoGrid.

5. Перестраиваем проект (в главном меню пункт **ПОСТРОЕНИЕ** - **Перестроить *Название проекта***).

## Введение

Терминология, необходимая для работы с ReoGrid, во многом аналогична таковой у Microsoft Excel. Здесь также присутствуют: **Книги (Workbooks)**, **Листы (Worksheets)**, **ячейки (Cells)**. В ячейках так же можно хранить как значения в основных форматах (текстовом, числовом, денежном, ...), так и формулы с использованием ряда функций, совместимых с английской и русской версиями Excel.

Экземпляр компонента (он же экземпляр класса **IWorkbook**) изначально содержит один **Лист**. Он доступен по псевдониму **Currentworksheet** (текущий лист).

Далее подразумевается, что в основную форму проекта (Windows Forms) добавлен экземпляр компонента **ReoGrid** с именем **rgrid**.

Новые **листы** книги могут создаваться как визуально - с использованием встроенного в **ReoGrid** вкладочного элемента (TabControl), так и программно - путём создания экземпляра объекта **WorkSheet** с последующим его добавлением в состав книги.

## Пора начинать

Рассмотрим основные моменты работы с компонентом на конкретных примерах. То есть, на основе рассмотренных ниже фрагментов кода можно собрать работоспособный редактор электронных таблиц. Изначально подразумевается, что на форму добавлен компонент ReoGrid с именем rgrid.

## Необходимые пространства имён (Namespaces)

Для создания редактора электронных таблиц будет достаточно подключить следующие пространства имён:

### VB

```
Imports System.IO
Imports System.Text
Imports unvell.ReoGrid 'Сам компонент
Imports unvell.ReoGrid.CellTypes 'Типы ячеек
Imports unvell.ReoGrid.Actions 'Действия
Imports unvell.ReoGrid.DataFormat 'Форматы данных
Imports unvell.ReoGrid.Print 'Печать
Imports unvell.ReoGrid.Chart 'Диаграммы
Imports unvell.ReoGrid.Events
```

### C#

```
using System.IO;
using System.Text;
using System;
using System.Windows.Forms;
using System.Drawing;
using System.Diagnostics;
using System.ComponentModel;
using unvell.ReoGrid; //Сам компонент
using unvell.ReoGrid.CellTypes; //Типы ячеек
using unvell.ReoGrid.Actions; //Действия
using unvell.ReoGrid.DataFormat; //Форматы данных
using unvell.ReoGrid.Print; //Печать
using unvell.ReoGrid.Chart; //Диаграммы
```

## Книга в памяти (виртуальная книга)

Помимо визуальной формы, ReoGrid может создавать виртуальную книгу ("книгу в памяти", "Memory workbook").

Виртуальная книга отличается от обычной отсутствием графического интерфейса и соответствующих свойств, событий, методов и действий, включая отмену/повтор действий. Требует гораздо меньше ресурсов компьютера, идеально подходит для выполнения некоторых фоновых задач, например: по полученным сведениям составить отчёт и сохранить его в виде файла.

### Возможности и ограничения виртуальных книг

#### Возможности:

- Работа с листами: создание, добавление, вставка, удаление, копирование, перемещение, поиск по имени (GetWorksheetByName) и по индексу (GetWorksheetIndex).
- Коллекция листов (Worksheets).
- Загрузка и сохранение данных: форматы RGF, Excel.
- Все основные события книги.
- Уведомления о возникших исключениях.

### Ограничения (функционал недоступен):

- Свойство **CurrentWorksheet**: у виртуальной книги нет визуального интерфейса, листа по умолчанию и переключателя листов. Управление листами происходит с помощью коллекции **Worksheets**.
- Методы и события, связанные с прокруткой (Scroll).
- Стили оформления самого компонента.
- Методы, связанные с действиями: **DoAction**, **Undo**, **Redo** и **RepeatLastAction**. Поэтому все действия выполняются мгновенно и без возможности отмены.

Пример работы с книгой в памяти приведён в конце данного руководства.

## Манипулирование листами книги

### VB

```
'Создание экземпляра виртуальной книги
Dim workbook = ReoGridControl.CreateMemoryWorkbook()

'Сброс книги (удаление всех листов и создание пустого нового)
rgrid.Reset()

'Создание экземпляра листа с именем по умолчанию
Dim sheet As Worksheet = rgrid.CreateWorksheet()
'или с указанным именем (имена должны быть уникальны)
Dim sheet1 As Worksheet = rgrid.CreateWorksheet("Мой_лист")

'Добавление листа в книгу в конец или с уточнением позиции
rgrid.AddWorksheet(sheet)
rgrid.InsertWorksheet(1, sheet)
'или
rgrid.Worksheets.Add(sheet)
rgrid.Worksheets.Insert(1, sheet)

'Определение количества листов в книге
Dim mycount As Integer = rgrid.Worksheets.Count

'Удаление листа с указанием экземпляра
rgrid.Worksheets.Remove(sheet)
'или
rgrid.RemoveWorksheet(sheet)
'или же с указанием позиции листа
rgrid.RemoveWorksheet(1)

'Копирование листа
Dim sheet2 As Worksheet = rgrid.CopyWorksheet(0, 1)

'Перемещение листа
rgrid.MoveWorksheet(0, 3)

'Сброс листа (удаление всех данных)
rgrid.CurrentWorksheet.Reset()

'Освобождение ресурсов, используемых листом
rgrid.Worksheets(0).Dispose()
```

### C#

```

//Создание экземпляра виртуальной книги
var workbook = ReoGridControl.CreateMemoryWorkbook();

//Сброс книги (удаление всех листов и создание пустого нового)
rgrid.Reset();

//Создание экземпляра листа с именем по умолчанию
Worksheet sheet = rgrid.CreateWorksheet();
//или с указанным именем (имена должны быть уникальны)
Worksheet sheet1 = rgrid.CreateWorksheet("Мой_лист");

//Добавление листа в книгу в конец или с уточнением позиции
rgrid.AddWorksheet(sheet);
rgrid.InsertWorksheet(1, sheet);
//или
rgrid.Worksheets.Add(sheet);
rgrid.Worksheets.Insert(1, sheet);

//Определение количества листов в книге
int mycount = rgrid.Worksheets.Count;

//Удаление листа с указанием экземпляра
rgrid.Worksheets.Remove(sheet);
//или
rgrid.RemoveWorksheet(sheet);
//или же с указанием позиции листа
rgrid.RemoveWorksheet(1);

//Копирование листа
Worksheet sheet2 = rgrid.CopyWorksheet(0, 1);

//Перемещение листа
rgrid.MoveWorksheet(0, 3);

//Сброс листа (удаление всех данных)
rgrid.CurrentWorksheet.Reset();

//Освобождение ресурсов, используемых листом
rgrid.Worksheets[0].Dispose();

```

## Основные (простые) действия над ячейками и диапазонами ячеек

### VB

```

'Занесение значения в ячейку
rgrid.CurrentWorksheet(1, 1) = "Какой-то текст"
rgrid.CurrentWorksheet("B2") = "Какой-то текст"

'Получение данных из ячейки (текст)
Dim strText As String = rgrid.CurrentWorksheet(1, 1).ToString
Dim strText1 As String = rgrid.CurrentWorksheet.Cells(1, 1).DisplayText

'Ввод и получение текстового представления формулы
rgrid.CurrentWorksheet.Cells(1, 1).Formula = "=A8+B8"
Dim strFormula As String = rgrid.CurrentWorksheet.Cells(1, 1).Formula

'Пример сброса типа ячейки к обычному (подробности ниже)
rgrid.CurrentWorksheet.Cells("E5").Body = Nothing

'Задание имени диапазона
Dim myRange = rgrid.CurrentWorksheet.DefineNamedRange("мой_диапазон", 3, 1, 5, 4)

'Запись данных в диапазон
myRange.Data = New Object(,) {{1, 2, 3, 4}, {0.1000000000000001, 0.2000000000000001, 0.299999999999999, 0.4000000000000002}},

'Задание ширины столбца
rgrid.CurrentWorksheet.SetColumnWidth(1, 4, 100)

'Изменяем ширину заголовка
rgrid.CurrentWorksheet.ColumnHeaders(0).Width = 100 'Столбец A
rgrid.CurrentWorksheet.ColumnHeaders(1).Width = 500 'Столбец B

```

### C#

```

//Занесение значения в ячейку
rgrid.CurrentWorksheet[1, 1] = "Какой-то текст";
rgrid.CurrentWorksheet["B2"] = "Какой-то текст";

//Получение данных из ячейки (текст)
string strText1 = rgrid.CurrentWorksheet.Cells[1, 1].DisplayText;

//Ввод и получение текстового представления формулы
rgrid.CurrentWorksheet.Cells[1, 1].Formula = "=A8+B8";
string strFormula = rgrid.CurrentWorksheet.Cells[1, 1].Formula;

//Пример сброса типа ячейки к обычному (подробности ниже)
rgrid.CurrentWorksheet.Cells["E5"].Body = null;

//Задание имени диапазона
var myRange = rgrid.CurrentWorksheet.DefineNamedRange("мой_диапазон", 3, 1, 5, 4);

//Запись данных в диапазон
myRange.Data = new object[, ] {{1,2,3,4},{0.1,0.2,0.3,0.4},{ "яблоко", "банан", "апельсин", "груша"}};

//Задание ширины столбца
rgrid.CurrentWorksheet.SetColumnWidth(1, 4, 100);

//Изменяем ширину заголовка
rgrid.CurrentWorksheet.ColumnHeaders[0].Width = 100; //Столбец A
rgrid.CurrentWorksheet.ColumnHeaders[1].Width = 500; //Столбец B

```

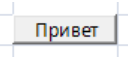

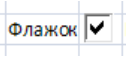
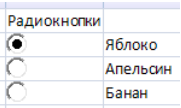
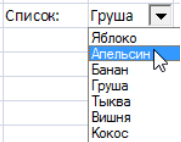



## Типы ячеек (Cell Types)

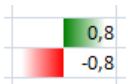
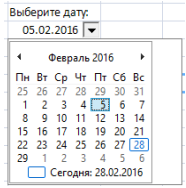
Для использования встроенных типов ячеек нужно импортировать соответствующее пространство имён: **unvell.ReoGrid.CellTypes**

Ячейки в **ReoGrid** могут иметь основу, унаследованную от **CellBody** или реализованную через интерфейс **ICellBody**.

Чтобы вернуть ячейке первоначальный облик, то есть вернуть возможность вводить данные и править текст, достаточно присвоить свойству **body** значение **null** (C#), **Nothing** (VB)

ReoGrid содержит следующие встроенные типы ячеек:

Название	Вид	Пояснение
Кнопка (Button)		Для каждой кнопки можно назначить свой обработчик событий.
Гиперссылка (Hyperlink)		По щелчку на ссылке в браузере по умолчанию открывается соответствующий сайт.
Флажок (Check Box)		Аналог флажка (CheckBox). Свойство для добавления поясняющего текста не предусмотрено.
Радиокнопка (Radio Button)		Аналог радиокнопки. Свойство для добавления поясняющего текста не предусмотрено.
Раскрывающийся список (Drop-down list)		Аналог классического раскрывающегося списка
Картинка (Image)		Изображение с настраиваемым расположением.
Индикатор выполнения (Numeric Progress)		Аналог ProgressBar. Значения: от 0 до 1
Кнопка с картинкой (ImageButtonCell)		Кнопка, для которой можно придать вид картинки.

<p>"Индикатор направления" (NegativeProgressCell)</p>		<p>В первая половина ячейки отвечает за отрицательные значения, вторая - за положительные. Рекомендуемые значения: интервал от -1 до 1</p>
<p>Выбор даты (DatePickerCell)</p>		<p>При щелчке на списке появляется встроенный элемент управления Календарь.</p>

Ниже приведены примеры кода для работы с указанными типами ячеек.

## Кнопка (Button)

### VB

```
'1. Самый простой вариант кнопки (Button)
rgrid.CurrentWorksheet(1, 1) = New ButtonCell()
rgrid.CurrentWorksheet(1, 1) = "Привет"

'2. Кнопка с обработчиком события нажатия (Click)
Dim btn As ButtonCell = New ButtonCell("Привет!")
rgrid.CurrentWorksheet(1, 1) = New Object() {"Кнопка: ", btn}

AddHandler btn.Click, AddressOf btn_Click
```

### C#

```
//1. Самый простой вариант кнопки (Button)
rgrid.CurrentWorksheet[1, 1] = new ButtonCell();
rgrid.CurrentWorksheet[1, 1] = "Привет";

//2. Кнопка с обработчиком события нажатия (Click)
ButtonCell btn = new ButtonCell("Привет!");
rgrid.CurrentWorksheet[1, 1] = new object[] {"Кнопка: ",btn};
```

Вариант обработчика события

### VB

```
Sub btn_Click(ByVal s As Object, ByVal e As EventArgs)
MsgBox("Кнопка нажата")
End Sub
```

### C#

```
var button = new ButtonCell("Привет");
button.Click += (s, e) => MessageBox.Show("Кнопка нажата");
```

## Гиперссылка (Hyperlink)

Гиперссылка использует данные ячейки как адрес URL.

### VB

```
'1. Делаем гиперссылкой выделенный диапазон.
rgrid.CurrentWorksheet(rgrid.CurrentWorksheet.SelectionRange) = New HyperlinkCell("https://reogrid.net/ru", False)

'2. Настройка всех основных параметров с последующим применением к нужной ячейке.
Dim MyLink As New HyperlinkCell
MyLink.LinkColor = Color.Blue
MyLink.ActivateColor = Color.Green
MyLink.VisitedColor = Color.Red
MyLink.LinkURL = "https://reogrid.net/ru"
MyLink.AutoNavigate = True
rgrid.CurrentWorksheet(1, 2) = MyLink

'Если по щелчку на гиперссылке должен происходить переход, то можно применить приведённый ниже код. Объявление переменной-гиперссы
Dim link = New HyperlinkCell("https://reogrid.net/ru")
rgrid.CurrentWorksheet(rgrid.CurrentWorksheet.SelectionRange) = New Object() {"Гиперссылка", link}

'AddHandler link.Click, AddressOf link_Click
'Sub link_Click(ByVal s, ByVal e)
'System.Diagnostics.Process.Start(rgrid.GetCellText(1, 2))
'End Sub
```

## C#

```
//1. Делаем гиперссылкой выделенный диапазон.
rgrid.CurrentWorksheet[rgrid.CurrentWorksheet.SelectionRange] = new HyperlinkCell("https://reogrid.net/ru", false);

//2. Настройка всех основных параметров с последующим применением к нужной ячейке.
HyperlinkCell MyLink = new HyperlinkCell();
MyLink.LinkColor = Color.Blue;
MyLink.ActivateColor = Color.Green;
MyLink.VisitedColor = Color.Red;
MyLink.LinkURL = "https://reogrid.net/ru";
MyLink.AutoNavigate = true;
rgrid.CurrentWorksheet[1, 2] = MyLink;

//Если по щелчку на гиперссылке должен происходить переход, то можно применить приведённый ниже код. Объявление переменной-гиперссы
var link = new HyperlinkCell("https://reogrid.net/ru");
rgrid.CurrentWorksheet[rgrid.CurrentWorksheet.SelectionRange] = new object[] {"Гиперссылка", link};
//link.Click += link_Click;
/*public void link_Click(s, e)
{
System.Diagnostics.Process.Start(rgrid.GetCellText(1, 2));
}*/
```

## Флажок (CheckBox)

Для этой процедуры рекомендуется использовать таблицу без объединённых ячеек ;-) Сам Флажок занимает одну ячейку, поясняющую надпись к нему нужно располагать с другой ячейке.

## VB

```
Dim middleStyle = New WorksheetRangeStyle
middleStyle.Flag = PlainStyleFlag.Padding
middleStyle.Flag = PlainStyleFlag.HorizontalAlign
middleStyle.Padding = New unvell.ReoGrid.PaddingValue(2)
middleStyle.HAlign = ReoGridHorAlign.Center
Dim grayTextStyle = New WorksheetRangeStyle
grayTextStyle.Flag = PlainStyleFlag.TextColor
grayTextStyle.TextColor = Color.DimGray

Dim checkbox As CheckBoxCell = New CheckBoxCell()
rgrid.CurrentWorksheet.SetRangeStyles(7, 2, 1, 1, middleStyle)
rgrid.CurrentWorksheet.SetRangeStyles(8, 2, 1, 1, grayTextStyle)
rgrid.CurrentWorksheet(7, 1) = New Object() {"Флажок", checkbox, "Напомнить через 5 минут"}
rgrid.CurrentWorksheet(8, 2) = "(Изменить состояние элемента управления можно и при помощи клавиатуры)"

'Пример создания обработчика. Для его работы может потребоваться объявление переменной checkbox на уровне класса
'AddHandler checkbox.CheckChanged, AddressOf checkbox_CheckChanged
'Sub checkbox_CheckChanged(ByVal sender, ByVal e)
'MsgBox("Флажок переведён в состояние " & checkbox.ButtonState.ToString())
'End Sub
```

## C#



```

var middleStyle = new WorksheetRangeStyle();
middleStyle.Flag = PlainStyleFlag.Padding;
middleStyle.Flag = PlainStyleFlag.HorizontalAlign;
middleStyle.Padding = new unvell.ReoGrid.PaddingValue(2);
middleStyle.HAlign = ReoGridHorAlign.Center;
dynamic grayTextStyle = new WorksheetRangeStyle();
grayTextStyle.Flag = PlainStyleFlag.TextColor;
grayTextStyle.TextColor = Color.DimGray;

CheckBoxCell checkbox = new CheckBoxCell();
rgrid.CurrentWorksheet.SetRangeStyles(7, 2, 1, 1, middleStyle);
rgrid.CurrentWorksheet.SetRangeStyles(8, 2, 1, 1, grayTextStyle);
rgrid.CurrentWorksheet[7, 1] = new object[] {"Флажок",checkbox, "Напомнить через 5 минут"};

rgrid.CurrentWorksheet[8, 2] = "(Изменить состояние элемента управления можно и при помощи клавиатуры)";
//Пример создания обработчика. Для его работы может потребоваться объявление переменной checkbox на уровне класса
//checkbox.CheckChanged += checkbox_CheckChanged;
/*public void checkbox_CheckChanged(sender, e)
{
Interaction.MsgBox("Флажок переведён в состояние " + checkbox.ButtonState.ToString());
}*/

```

## Радиокнопка (RadioButton)

Как и флажки, радиокнопки занимают одну ячейку. Назовём ячейку с радиокнопкой радиоячейкой. После добавления радиоячеек в группу, они начинают работать вместе (может быть выбрана только одна радиокнопка)

### VB

```

rgrid.CurrentWorksheet(0, 1) = "Радиокнопки"

'Группа
Dim radioGroup As RadioButtonGroup = New RadioButtonGroup()

'Объявление ячеек и подключение их к группе
Dim rr1 As RadioButtonCell = New RadioButtonCell
rr1.RadioGroup = radioGroup
Dim rr2 As RadioButtonCell = New RadioButtonCell
rr2.RadioGroup = radioGroup
Dim rr3 As RadioButtonCell = New RadioButtonCell
rr3.RadioGroup = radioGroup

'Применяем к таблице
rgrid.CurrentWorksheet(1, 1) = rr1
rgrid.CurrentWorksheet(2, 1) = rr2
rgrid.CurrentWorksheet(3, 1) = rr3

'Ну и добавляем пояснения для пользователя
rgrid.CurrentWorksheet(1, 2) = "Яблоко"
rgrid.CurrentWorksheet(2, 2) = "Апельсин"
rgrid.CurrentWorksheet(3, 2) = "Банан"

```

### C#

```

rgrid.CurrentWorksheet[0, 1] = "Радиокнопки";

//Группа
RadioButtonGroup radioGroup = new RadioButtonGroup();

//Объявление ячеек и подключение их к группе
RadioButtonCell rr1 = new RadioButtonCell();
rr1.RadioGroup = radioGroup;
RadioButtonCell rr2 = new RadioButtonCell();
rr2.RadioGroup = radioGroup;
RadioButtonCell rr3 = new RadioButtonCell();
rr3.RadioGroup = radioGroup;

//Применяем к таблице
rgrid.CurrentWorksheet[1, 1] = rr1;
rgrid.CurrentWorksheet[2, 1] = rr2;
rgrid.CurrentWorksheet[3, 1] = rr3;

//Ну и добавляем пояснения для пользователя
rgrid.CurrentWorksheet[1, 2] = "Яблоко";
rgrid.CurrentWorksheet[2, 2] = "Апельсин";
rgrid.CurrentWorksheet[3, 2] = "Банан";

```

## Раскрывающийся список (Drop-down List)

При выборе пункта он обновляет данные в своей ячейке. Получить выбранный пункт можно в обработчике события **CellDataChanged**:

### VB

```
Dim mydropdown As New DropDownListCell("Яблоко", "Апельсин", "Банан", "Груша", "Тыква", "Вишня", "Кокос")
rgrid.CurrentWorksheet(2, 3) = New Object() {"Список:", mydropdown}
```

### C#

```
DropDownListCell mydropdown = new DropDownListCell( "Яблоко", "Апельсин", "Банан", "Груша", "Тыква", "Вишня", "Кокос" );
rgrid.CurrentWorksheet[2, 3] = new object[] {"Список:", mydropdown};
```

## Картинка (Image)

В ячейку помещается графическое изображение

Свойство **ViewMode** позволяет управлять отображением картинки в ячейке:

### VB

```
'Загрузка из файла
Dim image1 As Image = Image.FromFile("D:\myinfo\photo.png")
rgrid.CurrentWorksheet(1, 2) = New ImageCell(image1)

'Получение из ресурсов
rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.photo)
'или
rgrid.CurrentWorksheet("B3") = New ImageCell(My.Resources.photo)

'Создаём ImageCell с указанием параметра отображения
Dim img As Image = Image.FromFile("D:\myinfo\photo.png")
Dim imgCell As ImageCell = New ImageCell(img, ImageCellViewMode.Zoom)
rgrid.CurrentWorksheet("B3") = imgCell

'Также можно указать значение напрямую
imgCell.ViewMode = ImageCellViewMode.Clip

'И даже можно задать параметры выравнивания по вертикали и горизонтали:
Dim cell = rgrid.CurrentWorksheet.Cells("F6")
cell.Style.HAlign = ReoGridHorAlign.Center
cell.Style.VAlign = ReoGridVerAlign.Middle

'Ещё варианты
rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.photo, ImageCellViewMode.Clip)
rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.photo, ImageCellViewMode.Stretch)
rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.photo, ImageCellViewMode.Zoom)
```

### C#

```
//Загрузка из файла
Image image1 = Image.FromFile("D:\myinfo\photo.png");
rgrid.CurrentWorksheet[1, 2] = new ImageCell(image1);

//Получение из ресурсов
rgrid.CurrentWorksheet[1, 2] = new ImageCell(reogrid_test_CSharp.Properties.Resources.photo);
//или
rgrid.CurrentWorksheet["B3"] = new ImageCell(reogrid_test_CSharp.Properties.Resources.photo);

//Создаём ImageCell с указанием параметра отображения
Image img = Image.FromFile("D:\myinfo\photo.png");
ImageCell imgCell = new ImageCell(img, ImageCellViewMode.Zoom);
rgrid.CurrentWorksheet["B3"] = imgCell;

//Также можно указать значение напрямую
imgCell.ViewMode = ImageCellViewMode.Clip;

//И даже можно задать параметры выравнивания по вертикали и горизонтали:
var cell = rgrid.CurrentWorksheet.Cells["F6"];
cell.Style.HAlign = ReoGridHorAlign.Center;
cell.Style.VAlign = ReoGridVerAlign.Middle;

//Ещё варианты
rgrid.CurrentWorksheet[1, 2] = new ImageCell(reogrid_test_CSharp.Properties.Resources.photo, ImageCellViewMode.Clip);
rgrid.CurrentWorksheet[1, 2] = new ImageCell(reogrid_test_CSharp.Properties.Resources.photo, ImageCellViewMode.Stretch);
rgrid.CurrentWorksheet[1, 2] = new ImageCell(reogrid_test_CSharp.Properties.Resources.photo, ImageCellViewMode.Zoom);
```

## Индикатор выполнения (Numeric Progress)

Позволяет отобразить результат выполнения какого-либо задания или его этапа.

### VB

```
Dim pc As ProgressCell = New ProgressCell
pc.TopColor = Color.Aqua
pc.BottomColor = Color.Black
rgrid.CurrentWorksheet(7, 2) = 0.8000000000000004

rgrid.CurrentWorksheet(7, 2) = pc
```

### C#

```
ProgressCell pc = new ProgressCell();
pc.TopColor = Color.Aqua;
pc.BottomColor = Color.Black;
rgrid.CurrentWorksheet[7, 2] = 0.8;

rgrid.CurrentWorksheet[7, 2] = pc;
```

## Кнопка с картинкой (ImageButtonCell)

### VB

```
Dim ibc As ImageButtonCell = New ImageButtonCell
ibc.Image = Image.FromFile("D:\myinfo\photo.png")
rgrid.CurrentWorksheet(9, 2) = 0.8000000000000004

rgrid.CurrentWorksheet(9, 2) = ibc
```

### C#

```
ImageButtonCell ibc = new ImageButtonCell();
ibc.Image = Image.FromFile("D:\myinfo\photo.png");
rgrid.CurrentWorksheet[9, 2] = 0.8;

rgrid.CurrentWorksheet[9, 2] = ibc;
```

## "Индикатор направления" (NegativeProgressCell)

### VB

```
Dim npc As NegativeProgressCell = New NegativeProgressCell
npc.NegativeColor = Color.Red
npc.LinearGradient = True
npc.PositiveColor = Color.Green
rgrid.CurrentWorksheet(10, 2) = 0.8000000000000004
rgrid.CurrentWorksheet(10, 2) = npc

Dim npc1 As NegativeProgressCell = New NegativeProgressCell
npc1.NegativeColor = Color.Red
npc1.LinearGradient = True
npc1.PositiveColor = Color.Green
rgrid.CurrentWorksheet(11, 2) = -0.8000000000000004
rgrid.CurrentWorksheet(11, 2) = npc1
```

### C#

```
NegativeProgressCell npc = new NegativeProgressCell();
npc.NegativeColor = Color.Red;
npc.LinearGradient = true;
npc.PositiveColor = Color.Green;
rgrid.CurrentWorksheet[10, 2] = 0.8;
rgrid.CurrentWorksheet[10, 2] = npc;

NegativeProgressCell npc1 = new NegativeProgressCell();
npc1.NegativeColor = Color.Red;
npc1.LinearGradient = true;
npc1.PositiveColor = Color.Green;
rgrid.CurrentWorksheet[11, 2] = -0.8;
rgrid.CurrentWorksheet[11, 2] = npc1;
```

## "Выбор даты" (DatePickerCell)

### VB

```
Dim myDatePicker As New DatePickerCell
rgrid.CurrentWorksheet(1, 3) = "Выберите дату:"
rgrid.CurrentWorksheet(2, 3) = New Object() {myDatePicker}
'или так:
rgrid.CurrentWorksheet(3, 3) = New Object() {"Выберите дату:", myDatePicker}
```

## C#

```
var myDatePicker = new DatePickerCell();
rgrid.CurrentWorksheet[1, 3] = "Выберите дату:";
rgrid.CurrentWorksheet[2, 3] = new Object[] {myDatePicker};
//или так:
rgrid.CurrentWorksheet[3, 3] = new Object[] {"Выберите дату:", myDatePicker};
```

# Форматы данных

Для использования форматов данных нужно импортировать пространство имён **unvell.ReoGrid.DataFormat**

Указываем формат данных ячейки с помощью метода листа **SetRangeDataFormat**:

## Встроенные шаблоны форматирования данных

Тип	CellDataFormatFlag	Аргумент
Числовой (Number)	CellDataFormatFlag.Number	NumberDataFormatter.NumberFormatArgs
Дата/Время (DateTime)	CellDataFormatFlag.DateTime	DateTimeDataFormatter.DateTimeFormatArgs
Процентный (Percent)	CellDataFormatFlag.Percent	PercentDataFormatter.PercentFormatArgs
Денежный (Currency)	CellDataFormatFlag.Currency	CurrencyDataFormatter.CurrencyFormatArgs
Текстовый (Text)	CellDataFormatFlag.Text	Нет
Настраиваемый (Custom Extension)	CellDataFormatFlag.Custom	Любой объект

Для числового типа доступно перечисление **NumberNegativeStyle**: стиль отображения отрицательных значений.

Стиль	Пример	Значение для перечисления
Без стиля	-123,456.789	Minus
Красный	123,456,789	Red
Красный цвет и минус	-123,456.789	Red   Minus (VB: Red Or Minus)
Помещение в скобки	(123,456.789)	Brackets
Помещение в скобки и красный цвет	(123,456.789)	Brackets   Red (VB: Brackets Or Red)
Префикс Sankaku	▲ 123,456,789	Prefix_Sankaku
Красный префикс Sankaku	▲ 123,456,789	Prefix_Sankaku   Red (VB: Prefix_Sankaku Or Red)

Пример использования:

## VB

```

'Числовой формат данных (Number)
Dim FFNum As New NumberDataFormatter.NumberFormatArgs
FFNum.DecimalPlaces = 4 'Количество знаков после запятой/точки: 0,1234
FFNum.NegativeStyle = NumberDataFormatter.NumberNegativeStyle.RedBrackets 'стиль отрицательных чисел
FFNum.UseSeparator = True 'Использовать разделитель: 123,456
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Number, FFNum)

'Дата/Время (DateTime)
Dim FFDT As New DateTimeDataFormatter.DateTimeFormatArgs
FFDT.CultureName = "ru-RU" 'Культура "ru-RU"
FFDT.Format = "dd/MM/yyyy" 'Шаблон
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.DateTime, FFDT)

'Процентный (Percent)
Dim FFPer As New NumberDataFormatter.NumberFormatArgs
FFPer.DecimalPlaces = 2 'Количество знаков после запятой или точки
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Percent, FFPer)

'Денежный (Currency)
Dim FFCur As New CurrencyDataFormatter.CurrencyFormatArgs
FFCur.CultureEnglishName = "ru-RU" 'Культура "ru-RU"
FFCur.DecimalPlaces = 1 'Количество знаков после запятой/точки
FFCur.PostfixSymbol = "p." 'Символ "p.". Указываем после числа
'FFCur.PrefixSymbol = "p." 'Символ "p.". Указываем перед числом
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Currency, FFCur)

'Текстовый (Text)
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Text, Nothing)

```

## C#

```

//Числовой формат данных (Number)
NumberDataFormatter.NumberFormatArgs FFNum = new NumberDataFormatter.NumberFormatArgs();
FFNum.DecimalPlaces = 4; //Количество знаков после запятой/точки: 0,1234
FFNum.NegativeStyle = NumberDataFormatter.NumberNegativeStyle.RedBrackets; //стиль отрицательных чисел
FFNum.UseSeparator = true; //Использовать разделитель: 123,456
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Number, FFNum);

//Дата/Время (DateTime)
DateTimeDataFormatter.DateTimeFormatArgs FFDT = new DateTimeDataFormatter.DateTimeFormatArgs();
FFDT.CultureName = "ru-RU"; //Культура "ru-RU"
FFDT.Format = "dd/MM/yyyy"; //Шаблон
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.DateTime, FFDT);

//Процентный (Percent)
NumberDataFormatter.NumberFormatArgs FFPer = new NumberDataFormatter.NumberFormatArgs();
FFPer.DecimalPlaces = 2; //Количество знаков после запятой или точки
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Percent, FFPer);

//Денежный (Currency)
CurrencyDataFormatter.CurrencyFormatArgs FFCur = new CurrencyDataFormatter.CurrencyFormatArgs();
FFCur.CultureEnglishName = "ru-RU"; //Культура "ru-RU"
FFCur.DecimalPlaces = 1; //Количество знаков после запятой/точки
FFCur.PostfixSymbol = "p."; //Символ "p.". Указываем после числа
//FFCur.PrefixSymbol = "p."; //Символ "p.". Указываем перед числом
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Currency, FFCur);

//Текстовый (Text)
rgrid.CurrentWorksheet.SetRangeDataFormat(RangePosition.EntireRange, CellDataFormatFlag.Text, null);

```

## Управление параметрами книги или листа

Существуют параметры для всей книги и для отдельных её листов. Для управления параметром существует три метода: универсальный (нужно указать параметр типа **WorkbookSettings** для книги, **WorksheetSettings** для листа, а также булево значение: True = включить, False = выключить), включение и выключение. Для последних вариантов в скобках нужно указать включаемый или отключаемый параметр типа **WorkbookSettings** или **WorksheetSettings**.

Узнать значение параметра можно с помощью метода **HasSetting**

### Параметры книги (WorkbookSettings) и листа (WorksheetSettings)

Варианты включения и выключения параметров, получение состояния параметра, а также комбинирование параметров. В качестве примера взят параметр `WorkbookSettings.View_ShowSheetTabControl`

## VB

```
'1. Книга
rgrid.SetSettings(WorkbookSettings.View_ShowSheetTabControl, True) 'Универсальный (True - включение, False - выключение)
rgrid.EnableSettings(WorkbookSettings.View_ShowSheetTabControl) 'Включение
rgrid.DisableSettings(WorkbookSettings.View_ShowSheetTabControl) 'Выключение
Dim bBook As Boolean = rgrid.HasSettings(WorkbookSettings.View_ShowSheetTabControl) 'Получаем текущее значение параметра

'2. Лист
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_Default, True) 'Универсальный вариант
rgrid.CurrentWorksheet.EnableSettings(WorksheetSettings.Behavior_Default) 'Включение
rgrid.CurrentWorksheet.DisableSettings(WorksheetSettings.Behavior_Default) 'Выключение
Dim bSheet As Boolean = rgrid.CurrentWorksheet.HasSettings(WorksheetSettings.Behavior_Default) 'Получаем текущее значение параметра
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_Default Or WorksheetSettings.Edit_ReadOnly, False) 'Выключение сразу
```

## C#

```
//1. Книга
rgrid.SetSettings(WorkbookSettings.View_ShowSheetTabControl, true); //Универсальный (true - включение, false - выключение)
rgrid.EnableSettings(WorkbookSettings.View_ShowSheetTabControl); //Включение
rgrid.DisableSettings(WorkbookSettings.View_ShowSheetTabControl); //Выключение
bool bBook = rgrid.HasSettings(WorkbookSettings.View_ShowSheetTabControl); //Получаем текущее значение параметра

//2. Лист
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_Default, true); //Универсальный вариант (true - включение, false -
rgrid.CurrentWorksheet.EnableSettings(WorksheetSettings.Behavior_Default); //Включение
rgrid.CurrentWorksheet.DisableSettings(WorksheetSettings.Behavior_Default); //Выключение
bool bSheet = rgrid.CurrentWorksheet.HasSettings(WorksheetSettings.Behavior_Default); //Получаем текущее значение параметра
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_Default | WorksheetSettings.Edit_ReadOnly, false); //Выключение сразу
```

## Пример настройки книги и листа с пояснениями в комментариях

## VB

```
'Включаем все параметры книги
rgrid.SetSettings(WorkbookSettings.View_ShowSheetTabControl, True) 'Показывать вкладки-листы
rgrid.SetSettings(WorkbookSettings.View_ShowScrolls, True) 'Показывать полосы прокрутки
rgrid.SetSettings(WorkbookSettings.View_ShowHorScroll, True) 'Показывать горизонтальную полосу прокрутки
rgrid.SetSettings(WorkbookSettings.View_ShowVerScroll, True) 'Показывать вертикальную полосу прокрутки

'Включаем все параметры листа
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_MouseWheelToScroll, True) 'Разрешить прокрутку таблицы колесом мыши
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_MouseWheelToZoom, True) 'Разрешить пользователю масштабировать табли
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_ShortcutKeyToZoom, True) 'Разрешить пользователю масштабировать табл
'rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_DragToMoveColumnHeader, True) 'Разрешить пользователю перемещать ве
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_ScrollToFocusCell, True) 'Всегда разрешать автоматическую прокрутку
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_AllowUserChangingPageBreaks, True) 'Разрешить пользователю вставлять

'Правка
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_Readonly, True) 'Элемент управления будет работать в режиме "Только для
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoFormatCell, True) 'Разрешить форматировать данные после правки текст
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_FriendlyPercentInput, True) 'Разрешить отображение сомвола "процент" при
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoExpandRowHeight, True) 'Разрешить настройку высоты строк при увеличе
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoExpandColumnWidth, True) 'Разрешить настройку ширины столбцов при ув
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustRowHeight, True) 'Разрешить пользователю настраивать высоту с
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustColumnWidth, True) 'Разрешить пользователю настраивать длину
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustColumnWidth, True) 'Разрешить пользователю регулировать ширин
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_DragSelectionToMoveCells, True) 'Разрешить перемещать или копировать выд

'Отображение, внешний вид
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowColumnHeader, True) 'Показывать заголовки столбцов
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowRowHeader, True) 'Показывать заголовки строк
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowHorizontalRuler, True) 'Показывать горизонтальную линейку (зарезерви
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowVerticalRuler, True) 'Показывать вертикальную линейку (зарезервирова
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowGridLine, True) 'Показывать линии сетки
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowHiddenCellLine, True) 'Разрешить отображение линии на скрытом заголо
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowShowRowOutlines, True) 'Разрешить показывать внешние границы строк,
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowShowColumnOutlines, True) 'Разрешить показ внешних границ столбцов,
'rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_MouseWheelToScroll, True) 'Разрешить прокрутку листа колесом мыши
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowPageBreaks, True) 'Разрешить отображать линии разделителя страниц
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowCellTextOverflow, True) 'Сделать текст ячеек отображаемым только вн
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowFrozenLine, True) 'Отображать или скрывать линию фиксации (frozen li

'Подсчёт формул
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Formula_AutoUpdateReferenceCell, True) 'Разрешить автоматическое обновление с
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Formula_AutoPickingCellAddress, True) 'Разрешить выбирать адреса выделенных я
```

C#

```

//Включаем все параметры книги
rgrid.SetSettings(WorkbookSettings.View_ShowSheetTabControl, true); //Показывать вкладки-листы
rgrid.SetSettings(WorkbookSettings.View_ShowScrolls, true); //Показывать полосы прокрутки
rgrid.SetSettings(WorkbookSettings.View_ShowHorScroll, true); //Показывать горизонтальную полосу прокрутки
rgrid.SetSettings(WorkbookSettings.View_ShowVerScroll, true); //Показывать вертикальную полосу прокрутки

//Включаем все параметры листа
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_MouseWheelToScroll, true); //Разрешить прокрутку таблицы колесом мыши
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_MouseWheelToZoom, true); //Разрешить пользователю масштабировать таб
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_ShortcutKeyToZoom, true); //Разрешить пользователю масштабировать та
//rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_DragToMoveColumnHeader, true); //Разрешить пользователю перемещать
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_ScrollToFocusCell, true); //Всегда разрешать автоматическую прокрутку
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Behavior_AllowUserChangingPageBreaks, true); //Разрешить пользователю вставля

//Правка
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_ReadOnly, true); //Элемент управления будет работать в режиме "Только дл
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoFormatCell, true); //Разрешить форматировать данные после правки тек
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_FriendlyPercentInput, true); //Разрешить отображение символа "процент" п
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoExpandRowHeight, true); //Разрешить настройку высоты строк при увели
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AutoExpandColumnWidth, true); //Разрешить настройку ширины столбцов при
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustRowHeight, true); //Разрешить пользователю настраивать высоту
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustColumnWidth, true); //Разрешить пользователю настраивать длин
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_AllowAdjustColumnWidth, true); //Разрешить пользователю регулировать шир
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Edit_DragSelectionToMoveCells, true); //Разрешить перемещать или копировать в

//Отображение, внешний вид
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowColumnHeader, true); //Показывать заголовки столбцов
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowRowHeader, true); //Показывать заголовки строк
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowHorizontalRuler, true); //Показывать горизонтальную линейку (зарезерв
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowVerticalRuler, true); //Показывать вертикальную линейку (зарезервиро
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowGridLine, true); //Показывать линии сетки
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowHiddenCellLine, true); //Разрешить отображение линии на скрытом заго
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowShowRowOutlines, true); //Разрешить показывать внешние границы стро
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowShowColumnOutlines, true); //Разрешить показывать внешние границы столбцо
//rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_MouseWheelToScroll, true); //Разрешить прокрутку листа колесом мыши
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowPageBreaks, true); //Разрешить отображать линии разделителя страниц
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_AllowCellTextOverflow, true); //Сделать текст ячеек отображаемым только
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.View_ShowFrozenLine, true); //Отображать или скрывать линию фиксации (frozen

//Подсчёт формул
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Formula_AutoUpdateReferenceCell, true); //Разрешить автоматическое обновление
rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Formula_AutoPickingCellAddress, true); //Разрешить выбирать адреса выделенных
//rgrid.CurrentWorksheet.SetSettings(WorksheetSettings.Formula_AutoRebuild, true); //Разрешить автоматически корректировать или пе

```

Параметр, который позволяет указать: нужно ли добавлять в конец листа немного свободного пространства (отступ от нижнего края листа):

**VB**

```
rgrid.ShowScrollEndSpacing = False
```

**C#**

```
rgrid.ShowScrollEndSpacing = false;
```

## Обработка событий

### События листа

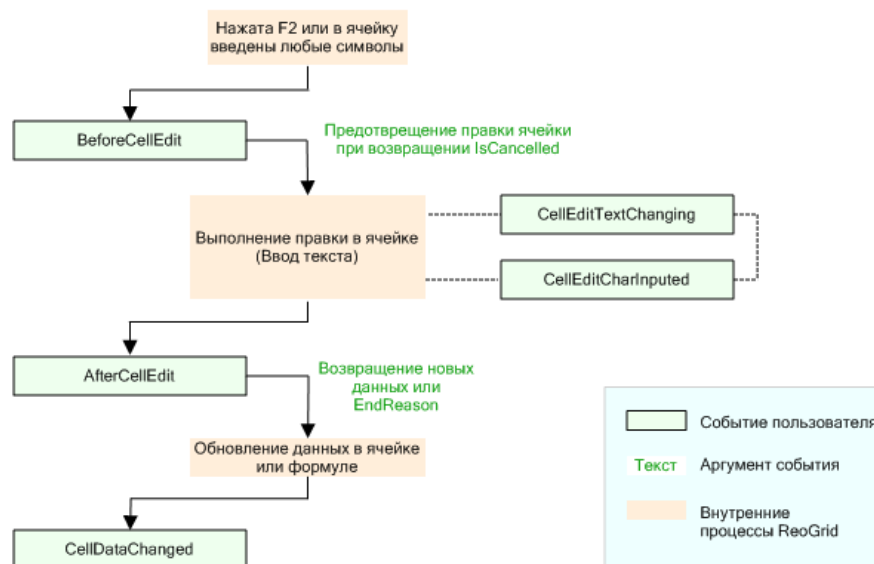
Событие	Когда возникает
WorksheetCreated	После создания листа
WorksheetInserted	После вставки листа
WorksheetRemoved	После удаления листа
BeforeWorksheetNameChange	Перед переименованием листа
WorksheetNameChanged	После переименования листа



## События ячеек

Событие	Когда возникает
BeforeCellEdit	Перед любыми изменениями в ячейках в режиме правки
AfterCellEdit	Когда какие-либо ячейки правятся пользователем
CellDataChanged	Когда изменяются данные в ячейках
CellKeyUp	Когда пользователь отпускает любую клавишу при активном листе внутри любой ячейки
CellMouseEnter	Когда указатель мыши перемещается в ячейку и зависает над ней
CellMouseLeave	Когда указатель мыши покидает ячейку
CellMouseDown	Когда происходит нажатие кнопки мыши, а указатель находится в ячейке
CellMouseUp	Когда происходит отпускание кнопки мыши, а указатель находится в ячейке
CellMouseMove	Когда указатель мыши движется внутри ячейки

### Принцип действия процедуры правки ячеек



Когда пользователь нажимает клавишу F2 или вводит любые символы (включая введённые инструментом IME), ReoGrid останавливает остальные процессы и переходит в режим правки ячейки (Edit mode), перед отображением на экране поля ввода ReoGrid вызовет возникновение события **BeforeCellEdit**. Это событие имеет аргумент **IsCancelled**, который применяется для отмены правки. Если правка разрешена, то поле ввода будет отображено на экране и получит фокус ввода.

Во время ввода пользователем любого текста в поле ввода, в ReoGrid будут происходить события **CellEditTextChanging** и **CellEditCharInputted**, при этом событие **CellEditTextChanging** будет возникать когда пользователь изменяет текст в поле ввода. При этом нет разницы в том, как вводится текст: путём прямого ввода, методом копирования/вставки или же другими путями. Событие **CellEditCharInputted** возникает при получении от пользователя символов, включая символы Unicode из инструмента IME. Это событие имеет аргументы позволяющие изменять вводимые символы или запретить их получение.

В WPF-издании, событие **CellEditCharInputted** можно проверять вводимые символы, но не изменять их. Для изменения текста нужно использовать событие **CellEditTextChanging**.

После того, как пользователь нажал клавишу Enter или убрал фокус ввода из поля ввода, процесс правки прерывается. По окончании правки возникают события **AfterCellEdit** и **CellDataChanged**. Обратите внимание на то, что когда пользователь нажмёт клавишу Escape, процесс правки будет отменён самим ReoGrid и введённые пользователем данные будут заменены теми данными, которые были в ячейке до правки, при этом произойдёт событие **AfterCellEdit**, а **CellDataChanged** не произойдёт.

### AfterCellEdit против CellDataChanged

**AfterCellEdit** предоставляет шанс прервать или отменить правку после того, как пользователь завершил ввод текста. Возвращение **EndReason** со значением **Cancel** из **AfterCellEditEventArgs** приведёт к тому, что правка будет прервана, а данные в ячейке заменятся теми, которые там были до начала правки. Событие **CellDataChanged** возникает тогда, когда данные были изменены каким-либо образом, включая метод **SetCellData** и вставку из Буфера обмена, это последнее событие, наблюдаемое после изменения данных в ячейке, оно не позволяет отменить правку или что-либо изменить.

## Подробнее о событии **CellKeyUp**

Пример использования:

### VB

```
'Добавление обработчика:
Dim sheet = rgrid.CurrentWorksheet
AddHandler sheet.CellKeyUp, AddressOf sheet_CellKeyUp
```

### C#

```
//Добавление обработчика:
var sheet = rgrid.CurrentWorksheet;
sheet.CellKeyUp += sheet_CellKeyUp;
```

Код обработчика

### VB

```
Sub sheet_CellKeyUp(ByVal Sender As Object, ByVal e As CellKeyDownEventArgs)
    MessageBox.Show("Нажата клавиша: " & e.KeyCode)
End Sub
```

### C#

```
void sheet_CellKeyUp(object sender, unvell.ReoGrid.Events.CellKeyDownEventArgs e){
    MessageBox.Show("Нажата клавиша: " + e.KeyCode);
}
```

**Примечание.** Это событие может не происходить, если были нажаты клавиши, переводящие лист в режим редактирования (Edit-mode). Обычно к ним относят символьные клавиши A-Z, 0-9 и подобные. Поэтому, если для ячеек разрешено редактирование, то могут произойти лишь события **BeforeCellKeyDown** и **AfterCellKeyDown**.

## Подробнее о событии **CellMouseDown**

Событие происходит при нажатии кнопки мыши, когда указатель мыши находится над видимой ячейкой.

**Примечание.** Значение свойства **Cell** аргумента события может быть равно **null** (VB: **Nothing**). ReoGrid не создаёт никаких экземпляров ячеек, если в ячейку не добавлялись данные или к ней не применялись стили.

Для того, чтобы всегда получать экземпляр ненулевой ячейки, используйте метод листа **CreateAndGetCell**.

### VB

```
'Добавление обработчика:
'Dim sheet = rgrid.CurrentWorksheet
'AddHandler sheet.CellMouseDown, AddressOf sheet_CellMouseDown

'Код обработчика
'Sub sheet_CellMouseDown(ByVal Sender As Object, ByVal e As CellMouseEventArgs)
'небезопасно: экземпляр ячейки может быть равным Nothing
'Dim cell = e.Cell

'безопасно: если экземпляр ячейки не существует, то он создаётся по информации о позиции
'Dim sheet = rgrid.CurrentWorksheet
'Dim cell = sheet.CreateAndGetCell(e.CellPosition)
'End Sub
```

### C#

```
//Добавление обработчика:
var sheet = rgrid.CurrentWorksheet;
//sheet.CellMouseDown += sheet_CellMouseDown;

//Код обработчика:
//void sheet_CellMouseDown(object sender, CellMouseEventArgs e) {
//небезопасно: экземпляр ячейки может быть равным null
//var cell = e.Cell;

//безопасно: если экземпляр ячейки не существует, то он создаётся по информации о позиции
//var sheet = rgrid.CurrentWorksheet;
//var cell = sheet.CreateAndGetCell(e.CellPosition);
//}
```

## Подробнее о событии **AfterCellEdit**

Событие возникает после того, как пользователь закончил редактировать содержимое ячейки.

Пример: с помощью обработки события **AfterCellEdit** изменяем текст в ячейке: из 'текст' получаем '[текст]':

## VB

```
'Добавление обработчика:  
'AddHandler rgrid.AfterCellEdit, AddressOf rgrid_AfterCellEdit  
  
'Код обработчика:  
'Sub rgrid_AfterCellEdit(ByVal Sender As Object, ByVal e As CellAfterEditEventArgs)  
'e.NewData = "[" & e.NewData & "]"  
'End Sub
```

## C#

```
//Добавление обработчика:  
//this.rgrid.AfterCellEdit += rgrid_AfterCellEdit;  
  
//Код обработчика:  
//void rgrid_AfterCellEdit(object sender, CellAfterEditEventArgs e)  
//{  
// e.NewData = "[" + e.NewData + "];  
//}
```

## События строк и столбцов

Событие	Когда возникает
RowInserted	Когда пользователь вставляет строки
RowDeleted	Когда пользователь удаляет строки
ColInserted	Когда пользователь вставляет столбцы
ColDeleted	Когда пользователь вставляет столбцы
RowsHeightChanged	Когда меняется высота строки
ColumnsWidthChanged	Когда меняется ширина столбца
HeightChanged	Когда меняется высота
WidthChanged	Когда меняется ширина
RowsFiltered	Когда к строкам применяется фильтр
RowsSorted	Когда к строкам применяется сортировка

## События диапазонов ячеек

Событие	Когда возникает
RangeMerged	Когда объединяются ячейки диапазона
RangeUnmerged	Когда отменяется объединение ячеек диапазона
RangeStyleChanged	Когда происходит установка стиля
BeforeRangeCopy	Перед копированием выделенного диапазона
BeforeRangeMove	Перед перемещением выделенного диапазона
AfterRangeCopy	После копирования диапазона
AfterRangeMove	После перемещения диапазона

## События границ

Событие	Когда возникает
BorderAdded	Когда добавляются границы
BorderRemoved	Когда удаляются границы

## События выделенного

Событие	Когда возникает
SelectionRangeChanged	При изменении диапазона выделения
SelectionRangeChanging	Во время изменения выделения мышью
SelectionModeChanged	При изменении режима выделения
SelectionStyleChanged	При изменении стиля выделения
SelectionForwardDirectionChanged	При изменении направления перемещения фокуса ввода
SelectionMovedForward	Когда выделение переносится в следующую позицию
HoverPosChanged	Когда указатель мыши движется над ячейками
FocusPosChanged	Когда фокус переходит к другой ячейке

## Подробнее о событии SelectionRangeChanged

Событие происходит при смене выделенного диапазона листа. Пример:

### VB

```
'Добавление обработчика:
'Dim sheet = rgrid.CurrentWorksheet
'AddHandler sheet.SelectionRangeChanged, AddressOf sheet_SelectionRangeChanged

'Код обработчика:
'Sub sheet_SelectionRangeChanged(byval sender as object, byval args as RangeEventArgs)
'MessageBox.Show("Выделен другой диапазон: " & args.Range.ToAddress())
'End Sub
```

### C#

```
//Добавление обработчика:
//var sheet = rgrid.CurrentWorksheet;
//sheet.SelectionRangeChanged += sheet_SelectionRangeChanged;

//Код обработчика:
//void sheet_SelectionRangeChanged(object sender, RangeEventArgs args)
//{
//    MessageBox.Show("Выделен другой диапазон: " + args.Range.ToAddress());
//}
```

**Примечание.** Событие не будет происходить во время перемещения мыши над листом перед выделением листа, происходит только когда кнопка мыши отпущена. Для получения информации об изменении выделенного диапазона во время перемещения мыши, используйте событие **SelectionRangeChanging**.

## События внешних границ (Outline)

Событие	Когда возникает
OutlineAdded	Когда в таблицу добавлена внешняя граница
OutlineRemoved	Когда внешняя граница удалена из таблицы
BeforeOutlineCollapse	Когда пользователь щёлкает на кнопке "-" за пределами таблицы - для её сворачивания
AfterOutlineCollapse	Когда граница замкнулась

BeforeOutlineExpand	Когда пользователь щёлкает на кнопке "+" за пределами таблицы - для её разворачивания
AfterOutlineExpand	Когда внешняя граница развёрнута

## События действий (Action)

Событие	Когда возникает
ActionPerformed	При выполнении любых действий
Undid	Когда действие undid
Redid	Когда действие redid

## События фиксации ("замораживания")

Событие	Когда возникает
CellsFrozen	Когда лист был зафиксирован
CellsUnfrozen	Когда прекращена фиксация листа

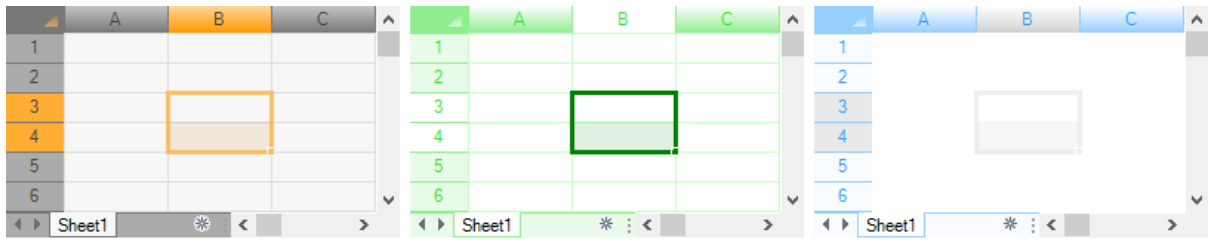
## События элемента управления

Событие	Когда возникает
Scaled	Во время масштабирования элемента управления (уменьшение/увеличение масштаба)
FileLoaded	Когда в элемент управления загрузились данные из файлового потока (load from given stream will not fire this event)
FileSaved	Когда содержимое элемента управления сохранено в файловом потоке (save into given stream will not fire this event)
Resetted	Когда состояние элемента управления сбрасывается к значению по умолчанию

## События буфера обмена

Событие	Когда возникает
BeforeCopy	Перед операцией копирования
AfterCopy	Во время копирования диапазона из буфера обмена (Clipboard)
BeforePaste	Перед операцией вставки
AfterPaste	Во время вставки из буфера обмена (Clipboard)
BeforeCut	Перед операцией вырезания
AfterCut	Когда пользователь вырезает содержимое диапазона
OnPasteError	При возникновении ошибок в процессе операции вставки

## Внешний вид компонента и локализация



## Смена цветовой палитры

Во ряде случаев оформление компонента по умолчанию может не подходить под оформление основного приложения. Для этого достаточно внести изменения в объект **ControlAppearanceStyle**. Наиболее удобным вариантом может быть встроенный в ReoGrid Editor редактор стилей, вызываемый из меню **Tools (Сервис) -> Control Appearance... (Оформление компонента)**. Полученный стиль можно экспортировать в набор строк на языках C#, Visual Basic, который необходимо вставить в код разрабатываемого приложения.

### VB

```
'Создание экземпляра стиля элемента управления, содержащего настройки цветов
Dim rgcs As ControlAppearanceStyle = New ControlAppearanceStyle(Color.Gray, Color.DarkOrange, False)

'Цвет текста будет серым
rgcs(ControlAppearanceColors.GridText) = Color.Gray
'Изменение других переменных...

'Применение обновлённого стиля
rgrid.ControlStyle = rgcs
```

### C#

```
// Создание экземпляра стиля элемента управления, содержащего настройки цветов
ControlAppearanceStyle rgcs = new ControlAppearanceStyle(Color.Gray, Color.DarkOrange, false);

// Цвет текста будет серым
rgcs[ControlAppearanceColors.GridText] = Color.Gray;
//Изменение других переменных...

// Применение обновлённого стиля
rgrid.ControlStyle = rgcs;
```

## Встроенная локализация

ReoGrid может переключать язык интерфейса автоматически - в зависимости от клиентской среды выполнения.

Модуль	Английский	Японский	Русский	Китайский
ReoGrid	✓	✓	✓	✓
ReoGridEditor	✓	✓	✓	
Demo Project	✓	✓		

## Локализация компонента

Для перевода надписей во встроенных элементах на нужный язык достаточно изменить значения нескольких свойств компонента.

### VB

```
'Локализация
unvell.ReoGrid.LanguageResource.Button_Cancel = "Отмена"
unvell.ReoGrid.LanguageResource.Button_OK = "OK"
unvell.ReoGrid.LanguageResource.Filter_SelectAll = "Выбрать все"
unvell.ReoGrid.LanguageResource.Filter_SortAtoZ = "Сортировка от А до Я"
unvell.ReoGrid.LanguageResource.Filter_SortZtoA = "Сортировка от Я до А"
unvell.ReoGrid.LanguageResource.Menu_DeleteSheet = "Удалить"
unvell.ReoGrid.LanguageResource.Menu_InsertSheet = "Вставить"
unvell.ReoGrid.LanguageResource.Menu_RenameSheet = "Переименовать"
unvell.ReoGrid.LanguageResource.Sheet = "Лист"
unvell.ReoGrid.LanguageResource.Sheet_RenameDialog_NameLabel = "Имя"
unvell.ReoGrid.LanguageResource.Sheet_RenameDialog_Title = "Переименование"

rgrid.CurrentWorksheet.Name = "Лист 1"
```

### C#

```
//Локализация
unvell.ReoGrid.LanguageResource.Button_Cancel = "Отмена";
unvell.ReoGrid.LanguageResource.Button_OK = "OK";
unvell.ReoGrid.LanguageResource.Filter_SelectAll = "Выбрать все";
unvell.ReoGrid.LanguageResource.Filter_SortAtoZ = "Сортировка от А до Я";
unvell.ReoGrid.LanguageResource.Filter_SortZtoA = "Сортировка от Я до А";
unvell.ReoGrid.LanguageResource.Menu_DeleteSheet = "Удалить";
unvell.ReoGrid.LanguageResource.Menu_InsertSheet = "Вставить";
unvell.ReoGrid.LanguageResource.Menu_RenameSheet = "Переименовать";
unvell.ReoGrid.LanguageResource.Sheet = "Лист";
unvell.ReoGrid.LanguageResource.Sheet_RenameDialog_NameLabel = "Имя";
unvell.ReoGrid.LanguageResource.Sheet_RenameDialog_Title = "Переименование";

rgrid.CurrentWorksheet.Name = "Лист 1";
```

## Инициализация списков

Все компоненты с префиксом **cbo**, упоминающиеся в данном руководстве, являются раскрывающимися списками **ComboBox**

## Заполняем списки перечнем кеглей (размеров) шрифта, гарнитур (семейств) шрифтов и количества процентов

### VB

```
'Кегли
cboFontSize.Items.AddRange({8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 28, 32, 36, 48, 60, 72, 96, 100})

'Гарнитуры
cboFontName.DataSource = FontFamily.Families
cboFontName.ValueMember = "Name"
cboFontName.Text = "Выберите гарнитуру"

'Масштабирование
cboZoom.Items.AddRange({"25%", "50%", "75%", "100%", "125%", "150%", "200%", "300%", "400%"})

'Объявление обработчиков событий изменения масштаба
'AddHandler rgrid.CurrentWorksheet.Scaled, AddressOf Sheet_Scaled
AddHandler cboZoom.TextChanged, AddressOf cboZoom_TextChanged
```

### C#

```
//Кегли
cboFontSize.Items.AddRange(new string[] { "8", "9", "10", "11", "12", "14", "16", "18", "20", "22", "24", "28", "32", "36", "48", "60", "72", "96", "100" });

//Гарнитуры
cboFontName.DataSource = FontFamily.Families;
cboFontName.ValueMember = "Name";
cboFontName.Text = "Выберите гарнитуру";

//Масштабирование
cboZoom.Items.AddRange(new string[] { "25%", "50%", "75%", "100%", "125%", "150%", "200%", "300%", "400%" });

//Объявление обработчика события изменения масштаба (уберите символы комментирования)
//rgrid.CurrentWorksheet.Scaled += Sheet_Scaled;
//cboZoom.TextChanged += cboZoom_TextChanged;
```

## Совместимость с форматом Excel

ReoGrid имеет повышенную совместимость с форматом Excel 2007/2010 (xlsx): возможно как считывание данных, так и их запись. При этом поддерживаются основные стили оформления ячеек, встроенные плавающие изображения и некоторые типы диаграмм. Созданный компонентом файл с расширением xlsx понимается и программой LibreOffice Calc.

ReoGrid представляет собой книгу (Workbook), которая может содержать вложенные листы-таблицы (Worksheets). Для переключения между листами служит встроенный отключаемый вкладочный элемент (TabControl). Активный лист имеет псевдоним **CurrentWorksheet**.

В примерах будут встречаться примеры загрузки и сохранения данных, используя формат Microsoft Excel. Наличие установленной копии офисного пакета от Майкрософт не требуется.

## Формула (Formula)

## Что следует учитывать при работе с формулами

1. Данные в ячейке, начинающиеся с символа "=", автоматически обрабатываются как формула.
2. Значения ячеек с формулами автоматически обновляются при изменении значений в связанных с ними ячейках.
3. Данные в ячейке, начинающиеся с символа (') не будут восприниматься как формула.
4. Язык формул **ReoGrid** чувствителен к регистру, все функции, совместимые с Microsoft Excel, должны указываться в **ВЕРХНЕМ** регистре, например **SUM** и **AVERAGE**, вызов функции и использованием имён вида "sum" приведёт к возникновению исключения 'function not found' (функция не найдена).

Можно использовать как английскую, так и русскую версии функций: они равнозначны.

## Поддерживаемые встроенные функции

Английская версия	Русская версия	Назначение функции
<b>Статистические функции</b>		
SUM	СУММ	Возвращает сумму значений чисел. Пример: SUM(B11:E12), СУММ(B11:E12)
AVERAGE	СРЗНАЧ	Возвращает среднее арифметическое значение указанных аргументов
COUNT	СЧЁТ	Возвращает количество ячеек, содержащих числа, и подсчитывает числа в списке аргументов
COUNTA	СЧЁТЗ	Функция СЧЁТЗ подсчитывает количество заполненных ячеек в диапазоне.
MIN	МИН	Возвращает наименьшее значение в списке аргументов
MAX	МАКС	Возвращает наибольшее значение в списке аргументов
<b>Функции условий</b>		
IF	ЕСЛИ	С проверкой на условие
AND	И	Логическое И
OR	ИЛИ	Логическое ИЛИ
NOT	НЕ	Логическое отрицание
<b>Адресные функции</b>		
ROW	СТРОКА	Возвращает номер строки, определяемой ссылкой.
COLUMN	СТОЛБЕЦ	Возвращает номер столбца для указанной ссылка на ячейку.
ADDRESS	АДРЕС	Возвращает ссылку на отдельную ячейку листа в виде текста
INDIRECT	ДВССЫЛ	Возвращает ссылку, заданную текстовой строкой
<b>Математические функции</b>		
ABS	ABS	Возвращает абсолютное значение числа
ROUND	ОКРУГЛ	Возвращает число, округленное с требуемой точностью. Примеры: ROUND(B8), ROUND(E8, 2)
CEILING	ОКРВВЕРХ	Возвращает число, округляемое в сторону увеличения, от нуля до ближайшего кратного указанному значению
FLOOR	ОКРВНИЗ	Округление числа до ближайшего меньшего значения
SIN	SIN	Возвращает синус числа
COS	COS	Возвращает косинус числа
TAN	TAN	Возвращает тангенс числа
ASIN	ASIN	Возвращает арксинус числа
ACOS	ACOS	Возвращает арккосинус числа
ATAN	ATAN	Возвращает арктангенс числа



ATAN2	ATAN2	Возвращает арктангенс для заданных координат x и y
POWER	СТЕПЕНЬ	Возвращает результат возведения числа в степень.
EXP	EXP	Возвращает число e, возведенное в указанную степень
LOG	LOG	Возвращает логарифм числа по заданному основанию.
LOG10	LOG10	Возвращает десятичный логарифм числа.
MOD	ОСТАТ	Возвращает остаток от деления аргумента "число" на значение аргумента "делитель". Результат имеет тот же знак, что и делитель.

#### Функции для работы с датами

NOW	ТДАТА	Возвращает текущие дату и время в числовом формате
TODAY	СЕГОДНЯ	Возвращает текущую дату в числовом формате
TIME	ВРЕМЯ	Возвращает заданное время в числовом формате
YEAR	ГОД	Преобразует дату в числовом формате в год
MONTH	МЕСЯЦ	Преобразует дату в числовом формате в месяцы
DAY	ДЕНЬ	Преобразует дату в числовом формате в день месяца
HOURL	ЧАС	Преобразует дату в числовом формате в часы
MINUTE	МИНУТЫ	Преобразует дату в числовом формате в минуты
SECOND	СЕКУНДЫ	Преобразует дату в числовом формате в секунды
MILLISECOND	MILLISECOND*	Преобразует дату в числовом формате в миллисекунды
DAYS	ДНЕЙ	?

#### Функции для работы со строками

LEFT	ЛЕВСИМВ	Возвращают крайние слева знаки текстового значения
RIGHT	ПРАВСИМВ	Возвращают крайние справа знаки текстовой строки
MID	ПСТР	Возвращают заданное число знаков из строки текста, начиная с указанной позиции
UPPER	ПРОПИСН	Преобразует все буквы текста в прописные
LOWER	СТРОЧН	Преобразует все буквы текста в строчные
LEN	ДЛСТР	Возвращает количество символов в текстовой строке
FIND	НАЙТИ	Поиск вхождения одного текстового значения в другое (с учетом регистра)
TRIM	СЖПРОБЕЛЫ	Удаляет из текста пробелы

#### Системные функции

ISERROR	ЕСЛИОШИБКА	Возвращает указанное значение, если вычисление по формуле вызывает ошибку; в противном случае функция возвращает результат формулы.
ISNUMBER	ЕЧИСЛО	Возвращает "True", если содержимое ячейки-аргумента является числом.

#### Дополнительные функции.

Английская версия	Русская версия	Назначение
SUMIF	СУММЕСЛИ	Возвращает сумму значений в диапазоне, которые соответствуют указанным критериям
AVERAGEIF	СРЗНАЧЕСЛИ	Возвращает среднее арифметическое значение указанных аргументов, которые отвечают единому заданному критерию
COUNTIF	СЧЁТЕСЛИ	Возвращает количество ячеек в заданном диапазоне, которые отвечают единому заданному критерию

VLOOKUP	ВПР	Позволяет найти одни данные через другие. Например: в таблице-справочнике найти название товара по его номеру.
---------	-----	--

\* Может быть изменено в будущих версиях

## Функции для строки формул (заполняем ComboBox с именем **cboFunctions**)

### VB

```
cboFunctions.Items.Add("SUM()")
cboFunctions.Items.Add("СУММ()")
cboFunctions.Items.Add("TODAY()")
cboFunctions.Items.Add("СЕГОДНЯ()")
```

### C#

```
cboFunctions.Items.Add("SUM()");
cboFunctions.Items.Add("СУММ()");
cboFunctions.Items.Add("TODAY()");
cboFunctions.Items.Add("СЕГОДНЯ()");
```

## Ссылки на ячейки (References of Cell)

Как и Microsoft Excel, ReoGrid поддерживает ссылки на ячейку и диапазон.

### VB

```
'Пример формулы со ссылками на ячейки (A1, B1) и диапазон (A1:C3)
rgrid.CurrentWorksheet("H8") = "=A1+B1-SUM(A1:C3)+AVERAGE(D1:H5)"

'Получение списка ссылок из ячейки
Dim rangeList = rgrid.CurrentWorksheet.GetCellFormulaReferenceRanges("H8")
```

### C#

```
//Пример формулы со ссылками на ячейки (A1, B1) и диапазон (A1:C3)
rgrid.CurrentWorksheet["H8"] = "=A1+B1-SUM(A1:C3)+AVERAGE(D1:H5)";

//Получение списка ссылок из ячейки
var rangeList = rgrid.CurrentWorksheet.GetCellFormulaReferenceRanges("H8");
```

## Трассировка прецедентов (Trace Precedents)

Для включения трассировки, необходимо установить для параметра ячейки **TraceFormulaPrecedent** значение **True**

При изменении формулы трассирующие стрелки исчезнут.

### VB

```
rgrid.CurrentWorksheet("C2") = 10
rgrid.CurrentWorksheet("C3") = 5
Dim cell As Cell = rgrid.CurrentWorksheet.Cells("C5")
cell.Formula = "=C2+C3"
cell.TraceFormulaPrecedents = True

'или путём вызова TraceCellPrecedents:
rgrid.CurrentWorksheet.TraceCellPrecedents("C5")

'Следующий код позволит постоянно отображать трассирующие стрелки
'Добавляем обработчик события
'AddHandler rgrid.CellDataChanged, AddressOf rgrid_CellDataChanged
'Sub rgrid_CellDataChanged(ByVal sender As Object, ByVal e As unvell.ReoGrid.Events.CellEventArgs)
'e.Cell.TraceFormulaPrecedents = True
'End Sub
```

### C#

```

rgrid.CurrentWorksheet["C2"] = 10;
rgrid.CurrentWorksheet["C3"] = 5;
Cell cell = rgrid.CurrentWorksheet.Cells["C5"];
cell.Formula = "=C2+C3";
cell.TraceFormulaPrecedents = true;

//или путём вызова TraceCellPrecedents:
rgrid.CurrentWorksheet.TraceCellPrecedents("C5");

//Следующий код позволит постоянно отображать трассирующие стрелки
//rgrid.CellDataChanged += (s, e) => e.Cell.TraceFormulaPrecedents = true;

```

## Трассировка зависимостей (Trace Dependents)

Определение именованного диапазона, установка значений именованного диапазона и применение формулы суммирования к именованному диапазону

### VB

```

Dim namedRange As NamedRange = rgrid.CurrentWorksheet.DefineNamedRange("items", "B2:E2")
namedRange.Data = New Object() {1, 3, 5, 7}
rgrid.CurrentWorksheet["F2"] = "=SUM(items)"

```

### C#

```

NamedRange namedRange = rgrid.CurrentWorksheet.DefineNamedRange("items", "B2:E2");
namedRange.Data = new object[] {1,3,5,7};
rgrid.CurrentWorksheet["F2"] = "=SUM(items)";

```

Переносим выбранную функцию из списка в текстовое поле формулы. Для этого используем обработчик события **SelectedIndexChanged** для раскрывающегося списка **cboFunctions**

### VB

```
txtFormula.Text &= cboFunctions.Text
```

### C#

```
txtFormula.Text += cboFunctions.Text;
```

Перенос формулы в выбранную ячейку. Используется обработчик события **KeyDown** для текстового поля формулы **txtFormula**

Изменение формулы в текстовом поле (обработка события **txtFormula\_KeyDown**)

### VB

```

Try
    If e.KeyCode = Keys.Enter Then
        rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).Formula = txtFormula.Text
    End If
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

### C#

```

if (e.KeyCode == Keys.Enter) {
    rgrid.CurrentWorksheet.Cells[rgrid.CurrentWorksheet.FocusPos].Formula = txtFormula.Text;
}

```

Получение текста формулы из выбранной пользователем ячейки. Для этого используется обработчик события **FocusPosChanged** компонента **ReoGrid**.

### VB

```

'Получаем значение поля Formula
Dim strFormula As String = rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).Formula

'Если поле не пустое - заносим его значение в строку формул, в поле списка функций заносим основные данные из ячейки
If Len(strFormula) >= 0 Then
    txtFormula.Text = strFormula
    cboFunctions.Text = rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).DisplayText
End If

```

### C#

```
//Получаем значение поля Formula
{
string strFormula = rgrid.CurrentWorksheet.Cells[rgrid.CurrentWorksheet.FocusPos].Formula;

//Если поле не пустое - заносим его значение в строку формул, в поле списка функций заносим основные данные из ячейки
if (strFormula.Length >= 0) {
txtFormula.Text = strFormula;
cboFunctions.Text = rgrid.CurrentWorksheet.Cells[rgrid.CurrentWorksheet.FocusPos].DisplayText;
}}

```

## Раздел "Файл" - функции, обычно доступные в меню "Файл"

### Сброс таблицы или листа ("Новый документ"). Очистка листа или книги

Если применить метод **Reset** к книге, то из неё удаляются все листы, после чего создаётся один пустой. Если же применить метод к листу - лист просто очистится от данных.

#### VB

```
rgrid.Reset()
rgrid.CurrentWorksheet.Reset()
```

#### C#

```
rgrid.Reset();
rgrid.CurrentWorksheet.Reset();
```

## Загрузка данных

Загрузка данных выполняется при помощи перегруженной функции **Load** (для всей книги) или по одному листу с использованием функций **LoadCSV**, **LoadRGF**, которые вполне могут исчезнуть в следующих версиях.

Первым её параметром может быть имя файла или файловый поток ввода-вывода

Второй параметр - формат открываемого файла из перечисления **IO.FileFormat** (CSV, Excel2007, ReoGridFormat, \_Auto - автовыбор, \_Custom - ещё не реализован)

Третий и необязательный параметр - кодировка символов. Его использование требует импортировать пространство имён **System.Text**

#### VB

```
'Загрузка данных из файла формата ReoGridFormat с использованием файлового потока
Using ss As FileStream = New FileStream(Application.StartupPath & "\test.rgf", FileMode.Create, FileAccess.Read)
    rgrid.Load(ss, IO.FileFormat.ReoGridFormat, Encoding.UTF8)
End Using
```

```
'Загрузка данных из файлов поддерживаемых форматов
'Вариант с функцией Load (книга целиком)
rgrid.Load(Application.StartupPath & "\test.csv", IO.FileFormat.CSV, Encoding.UTF8) 'CSV
rgrid.Load(Application.StartupPath & "\test.rgf", IO.FileFormat.ReoGridFormat, Encoding.UTF8) 'ReoGridFormat
rgrid.Load(Application.StartupPath & "\test.xlsx", IO.FileFormat.Excel2007, Encoding.UTF8) 'Excel 2007/2010
```

```
'Вариант с функцией LoadCSV (только один лист)
rgrid.CurrentWorksheet.LoadCSV(Application.StartupPath & "\test.csv", Encoding.UTF8)
```

#### C#

```
//Загрузка данных из файла формата ReoGridFormat с использованием файлового потока
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.rgf", FileMode.Create, FileAccess.Read)) {
    rgrid.Load(ss, unvell.ReoGrid.IO.FileFormat.ReoGridFormat, Encoding.UTF8);
}
```

```
//Загрузка данных из файлов поддерживаемых форматов
//Вариант с функцией Load (книга целиком)
rgrid.Load(Application.StartupPath + "\\test.csv", unvell.ReoGrid.IO.FileFormat.CSV, Encoding.UTF8); //CSV
rgrid.Load(Application.StartupPath + "\\test.rgf", unvell.ReoGrid.IO.FileFormat.ReoGridFormat, Encoding.UTF8); //ReoGridFormat
rgrid.Load(Application.StartupPath + "\\test.xlsx", unvell.ReoGrid.IO.FileFormat.Excel2007, Encoding.UTF8); //Excel 2007/2010
```

```
//Вариант с функцией LoadCSV (только один лист)
rgrid.CurrentWorksheet.LoadCSV(Application.StartupPath + "\\test.csv", Encoding.UTF8);
```

## Сохранение данных

Сохранение данных выполняется при помощи перегруженной функции **Save**. Её параметры аналогичны функции **Load**

Первым её параметром может быть имя файла или файловый поток ввода-вывода

Второй параметр - формат открываемого файла из перечисления **IO.FileFormat** (CSV, Excel2007, ReoGridFormat, \_Auto - автовыбор, \_Custom - ещё не реализован)

Третий и необязательный параметр - кодировка символов. Его использование требует импортировать пространство имён System.Text

Экспорт в формат HTML выполняется только через файловый поток и только для одного листа. Возможны два варианта: с заголовком (title) по умолчанию или же с произвольным заголовком

## VB

```
'Сохранение данных в файл формата ReoGridFormat с использованием файлового потока
Using ss As FileStream = New FileStream(Application.StartupPath & "\test.rgf", FileMode.Create, FileAccess.Write)
    rgrid.Save(ss, IO.FileFormat.ReoGridFormat, Encoding.UTF8)
End Using

'Сохранение с использованием методов
rgrid.CurrentWorksheet.ExportAsCSV(Application.StartupPath & "\test.csv", , Encoding.UTF8) 'CSV
rgrid.Save(Application.StartupPath & "\test.rgf", IO.FileFormat.ReoGridFormat, Encoding.UTF8) 'ReoGridFormat
rgrid.Save(Application.StartupPath & "\test.xlsx", IO.FileFormat.Excel2007, Encoding.UTF8) 'Excel 2007/2010

'HTML
Using ss As FileStream = New FileStream(Application.StartupPath & "\test.html", FileMode.Create, FileAccess.Write)
    rgrid.CurrentWorksheet.ExportAsHTML(ss) 'Заголовок по умолчанию
    rgrid.CurrentWorksheet.ExportAsHTML(ss, "Произвольный заголовок") 'Произвольный заголовок
End Using
```

## C#

```
//Сохранение данных в файл формата ReoGridFormat с использованием файлового потока
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.rgf", FileMode.Create, FileAccess.Write)) {
    rgrid.Save(ss, unvell.ReoGrid.IO.FileFormat.ReoGridFormat, Encoding.UTF8);
}

//Сохранение с использованием методов
rgrid.CurrentWorksheet.ExportAsCSV(Application.StartupPath + "\\test.csv", string.Empty, Encoding.UTF8); //CSV
rgrid.Save(Application.StartupPath + "\\test.rgf", unvell.ReoGrid.IO.FileFormat.ReoGridFormat, Encoding.UTF8); //ReoGridFormat
rgrid.Save(Application.StartupPath + "\\test.xlsx", unvell.ReoGrid.IO.FileFormat.Excel2007, Encoding.UTF8); //Excel 2007/2010

//HTML
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.html", FileMode.Create, FileAccess.Write)) {
    rgrid.CurrentWorksheet.ExportAsHTML(ss); //Заголовок по умолчанию
    rgrid.CurrentWorksheet.ExportAsHTML(ss, "Произвольный заголовок"); //Произвольный заголовок
}
```

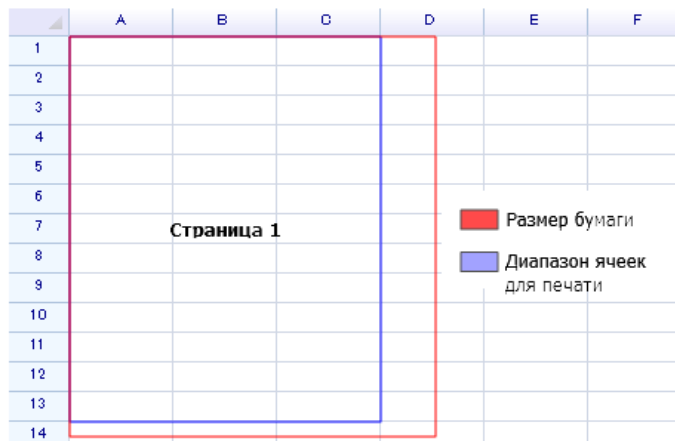
## Печать (Paging and Printing (<https://reogrid.net/document/paging-and-print/>))

### Авторазбивка на страницы (Auto Paging)

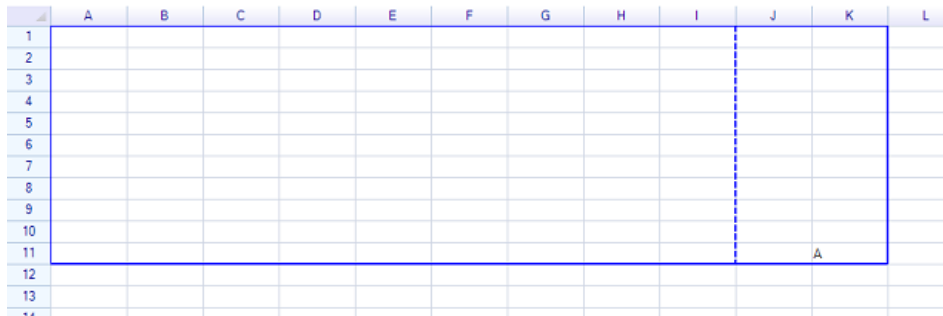
При выводе таблицы на печать, её содержимое автоматически разделяется между страницами.

	A	B	C	D	E	F	G	H
1								
2	(Разбивка по вертикали)							
3								
4								
5								
6								
7	Страница 1			Страница 3			Страница 5	
8								
9								
10								
11								
12								
13								
14								
15	Страница 2			Страница 4			Страница 6	
16								
17								

Как и в Excel, ReoGrid определяет, сколько ячеек можно разместить на каждой странице. Если все ячейки не помещаются на данной странице, то остаток переносится на следующую страницу.

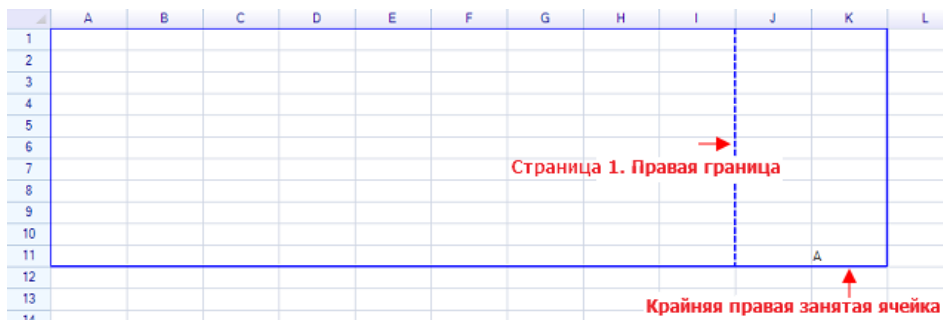


### Отображение линий разрывов страниц на экране



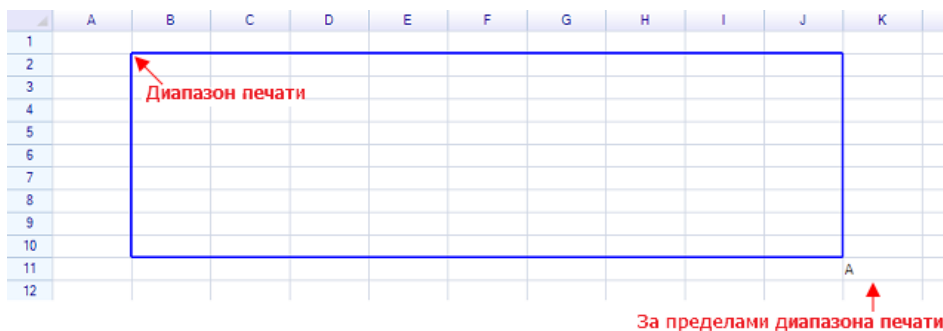
### Автопроверка диапазона печати

Если диапазон печати неопределён, то ReoGrid попытается найти максимум ячеек, заполненных данными и установит диапазон печати автоматически.



### Изменение диапазона печати

Для изменения диапазона печати листа используется свойство **PrintableRange**



Если размер диапазона печати превышает размер страницы, то он будет разбит автоматически. Ячейки за пределами диапазона печати будут распечатаны.

### Печать линий сетки таблицы

За это отвечает свойство **ShowGridLines**

### Изменение порядка следования страниц



Вниз и вправо



Вправо и вниз

## Изменение параметров страницы

Для этого используется свойство **PageSettings**

Теперь максимум данных из ячейки 'A' могут быть распечатаны в альбомной ориентации бумаги

## Вставка разрывов страниц

Для их вставки предназначены свойства **RowPageBreaks** и **ColumnPageBreaks**.

Обновление разбивки страницы выполняется уже знакомым методом **AutoSplitPage**.

Пример того, как таблица разбивается на две страницы:

## Системные и пользовательские разрывы страниц

Существует два вида разрывов страниц: обозначаемые пунктирной линией системные, получаемые при помощи метода **AutoSplitPage**, а также обозначаемые сплошной линией пользовательские, вставляемые или перетаскиваемые мышью пользователем вручную.

## Изменение разрывов страниц

Для изменения/перемещения индексов системных разрывов страниц используются следующие методы **ChangeColumnPageBreak** и **ChangeRowPageBreak**

После изменения системных индексов на пользовательские, пунктирная линия сменится сплошной.

## Повторения страниц (Iterate pages)

Для повторения всех печатаемых страниц используется метод **IteratePrintPage**, возвращающий диапазон печати для каждой страницы.

Пример результата вывода на принтер (см. пример ниже):

```
Диапазон печати: A1:E11  
Диапазон печати: F1:K11
```

По-возможности, повторение страниц (iterate pages) определяется порядком (синтаксис C#):

```
rgrid.CurrentWorksheet.IteratePrintPages(PrintPageOrder.OverThenDown, range => { ... });  
Если порядок следования страниц не будет указан, то будет использоваться порядок печати.
```

## Отключение возможности изменения/подстройки разрывов страниц пользователем

Для этого используется параметр настройки **WorksheetSettings.Behavior\_AllowUserChangingPageBreaks**

Основные действия показаны в следующих блоках кода:

### VB

```
'Авторазбивка на страницы  
rgrid.CurrentWorksheet.AutoSplitPage()  
  
'Отображение линий разрывов страниц на экране  
rgrid.CurrentWorksheet.EnableSettings(WorksheetSettings.View_ShowPageBreaks)  
  
'Изменение диапазона печати  
rgrid.CurrentWorksheet.PrintableRange = New RangePosition(1, 1, 9, 9)  
  
'Изменение порядка следования страниц  
rgrid.CurrentWorksheet.PrintSettings.PageOrder = PrintPageOrder.DownThenOver  
  
'Печать линий сетки таблицы  
rgrid.CurrentWorksheet.PrintSettings.ShowGridLines = True  
  
'Изменение параметров страницы  
'Для этого используется свойство PageSettings  
'rgrid.CurrentWorksheet.PageSettings.Landscape = True  
'rgrid.CurrentWorksheet.PrintSettings.Landscape = True  
  
'Вставка разрывов страниц  
'rgrid.CurrentWorksheet.RowPageBreaks.Add(5)  
'rgrid.CurrentWorksheet.ColumnPageBreaks.Add(5)  
  
'Обновление разбивки страницы  
rgrid.CurrentWorksheet.AutoSplitPage()  
  
'Изменение разрывов страниц  
rgrid.CurrentWorksheet.ChangeColumnPageBreak(0, 1) 'Старый индекс, Новый индекс  
rgrid.CurrentWorksheet.ChangeRowPageBreak(0, 1) ' Старый индекс, Новый индекс  
  
'Повторения страниц (Iterate pages)  
rgrid.CurrentWorksheet.IteratePrintPages(Function(range)  
    Console.WriteLine("Диапазон печати: " + range.ToAddress())  
    Return True  
End Function)  
  
'Отключение возможности изменения/подстройки разрывов страниц пользователем  
rgrid.CurrentWorksheet.DisableSettings(WorksheetSettings.Behavior_AllowUserChangingPageBreaks)
```

### C#



```

//Авторазбивка на страницы
rgrid.CurrentWorksheet.AutoSplitPage();

//Отображение линий разрывов страниц на экране
rgrid.CurrentWorksheet.EnableSettings(WorksheetSettings.View_ShowPageBreaks);

//Изменение диапазона печати
rgrid.CurrentWorksheet.PrintableRange = new RangePosition(1, 1, 9, 9);

//Изменение порядка следования страниц
rgrid.CurrentWorksheet.PrintSettings.PageOrder = PrintPageOrder.DownThenOver;

//Печать линий сетки таблицы
rgrid.CurrentWorksheet.PrintSettings.ShowGridLines = true;

//Изменение параметров страницы
//Для этого используется свойство PageSettings
//rgrid.CurrentWorksheet.PageSettings.Landscape = true;
//rgrid.CurrentWorksheet.PrintSettings.Landscape = true;

//Вставка разрывов страниц
//rgrid.CurrentWorksheet.RowPageBreaks.Add(5);
//rgrid.CurrentWorksheet.ColumnPageBreaks.Add(5);

//Обновление разбивки страницы
rgrid.CurrentWorksheet.AutoSplitPage();

//Изменение разрывов страниц
rgrid.CurrentWorksheet.ChangeColumnPageBreak(0, 1);//Старый индекс, Новый индекс
rgrid.CurrentWorksheet.ChangeRowPageBreak(0, 1);//Старый индекс, Новый индекс

//Повторения страниц (Iterate pages)
rgrid.CurrentWorksheet.IteratePrintPages(range =>
{
Console.WriteLine("Диапазон печати: " + range.ToAddress());
return true;
});

//Отключение возможности изменения/подстройки разрывов страниц пользователем
rgrid.CurrentWorksheet.DisableSettings(WorksheetSettings.Behavior_AllowUserChangingPageBreaks);

```

## Сессии печати (Print Session) и предварительный просмотр

Представляют собой сессии для вывода на печать определённых листов или целой книги.



VB

```

'Вывод на печать содержимого одного листа с настройками по умолчанию
Dim session = rgrid.CurrentWorksheet.CreatePrintSession()
session.Print()

'Вывод на печать содержимого книги (всех листов). Доступно только в Pro-издании
Dim session1 = rgrid.CreatePrintSession()
session1.Print()

'Вывод на печать определённых листов. Доступно только в Pro-издании
Dim session2 = rgrid.CreatePrintSession()
session2.Worksheets.Clear()
session2.Worksheets.Add(rgrid.Worksheets("Sheet1"))
session2.Worksheets.Add(rgrid.Worksheets("Sheet3"))
session2.Print()

'Предварительный просмотр (Print Preview). Отображение диалогового окна
Dim sheet = rgrid.CurrentWorksheet
Using session3 = sheet.CreatePrintSession()
    Using ppd As New PrintPreviewDialog()
ppd.Document = session3.PrintDocument
ppd.SetBounds(200, 200, 1024, 768)
ppd.PrintPreviewControl.Zoom = 1.0
ppd.ShowDialog(Me)
    End Using
End Using

```

## C#

```

//Вывод на печать содержимого одного листа с настройками по умолчанию
var session = rgrid.CurrentWorksheet.CreatePrintSession();
session.Print();

//Вывод на печать содержимого книги (всех листов). Доступно только в Pro-издании
var session1 = rgrid.CreatePrintSession();
session1.Print();

//Вывод на печать определённых листов. Доступно только в Pro-издании
var session2 = rgrid.CreatePrintSession();
session2.Worksheets.Clear();
session2.Worksheets.Add(rgrid.Worksheets["Sheet1"]);
session2.Worksheets.Add(rgrid.Worksheets["Sheet3"]);
session2.Print();

//Предварительный просмотр (Print Preview). Отображение диалогового окна
var sheet = rgrid.CurrentWorksheet;

using (var session3 = sheet.CreatePrintSession())
{
    using (PrintPreviewDialog ppd = new PrintPreviewDialog())
    {
ppd.Document = session3.PrintDocument;
ppd.SetBounds(200, 200, 1024, 768);
ppd.PrintPreviewControl.Zoom = 1d;
ppd.ShowDialog(this);
    }
}

```

## Отображение системного диалогового окна настройки параметров страницы

1. Создаём экземпляр диалогового окна
2. Импортируем настройки из параметров листа
3. Настраиваем параметры
4. Как только пользователь выбрал нужные настройки и нажал экранную кнопку "OK", импортируем настройки обратно в параметры листа ReoGrid

## VB

```

Using psd As PageSetupDialog = New PageSetupDialog()
    psd.PageSettings = rgrid.CurrentWorksheet.PrintSettings.CreateSystemPageSettings()

    psd.AllowMargins = True
    psd.AllowPrinter = True
    psd.AllowPaper = True
    psd.EnableMetric = True

    If psd.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
rgrid.CurrentWorksheet.PrintSettings.ApplySystemPageSettings(psd.PageSettings)
    End If
End Using

```

**C#**

```

using (PageSetupDialog psd = new PageSetupDialog())
{
psd.PageSettings = rgrid.CurrentWorksheet.PrintSettings.CreateSystemPageSettings();

    psd.AllowMargins = true;
    psd.AllowPrinter = true;
    psd.AllowPaper = true;
    psd.EnableMetric = true;

    if (psd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
rgrid.CurrentWorksheet.PrintSettings.ApplySystemPageSettings(psd.PageSettings);
    }
}

```

## Вариант вызова диалога предварительного просмотра

**VB**

```

'Создаём документ для печати
Dim docp As System.Drawing.Printing.PrintDocument = Nothing

'Показываем диалоговое окно предварительного просмотра
Using ppd As PrintPreviewDialog = New PrintPreviewDialog()
    ppd.Document = docp
    ppd.SetBounds(200, 200, 1024, 768)
    ppd.PrintPreviewControl.Zoom = 0.5D '1D
    ppd.PrintPreviewControl.Columns = 2
    ppd.PrintPreviewControl.Rows = 2
    ppd.ShowDialog(Me)
End Using

```

**C#**

```

//Создаём документ для печати
System.Drawing.Printing.PrintDocument docp = null;//rgrid.CurrentWorksheet.CreatePrintSession();

//Показываем диалоговое окно предварительного просмотра
using (PrintPreviewDialog ppd = new PrintPreviewDialog()) {
    ppd.Document = docp;
    ppd.SetBounds(200, 200, 1024, 768);
    ppd.PrintPreviewControl.Zoom = 0.5D;//1D
    ppd.PrintPreviewControl.Columns = 2;
    ppd.PrintPreviewControl.Rows = 2;
    ppd.ShowDialog(this);
}

```

## Вывод содержимого листа на печать с использованием стандартного диалогового окна печати

**VB**

```

Dim doc As System.Drawing.Printing.PrintDocument = Nothing
Try
    doc = rgrid.CurrentWorksheet.CreatePrintSession().PrintDocument

Catch ex As Exception
    MessageBox.Show(Me, ex.Message, Me.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Information)
Return
End Try

Using pd = New System.Windows.Forms.PrintDialog()
    pd.Document = doc
    pd.UseEXDialog = True 'в 64-битной Windows

    If pd.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
doc.PrinterSettings = pd.PrinterSettings
doc.Print()
    End If
End Using

If doc Is Nothing = False Then doc.Dispose()

```

## C#

```

System.Drawing.Printing.PrintDocument doc = null;

try {
    doc = rgrid.CurrentWorksheet.CreatePrintSession().PrintDocument;
}
catch (Exception ex)
{
    MessageBox.Show(this, ex.Message, this.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Information);
    return;
}

using (var pd = new System.Windows.Forms.PrintDialog())
{
    pd.Document = doc;
    pd.UseEXDialog = true; // в 64-битной Windows

    if (pd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        doc.PrinterSettings = pd.PrinterSettings;
        doc.Print();
    }
}

if (doc != null) doc.Dispose();

```

## Сброс разрывов страниц

### VB

```

Try
    rgrid.CurrentWorksheet.ResetAllPageBreaks()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

### C#

```

try {
    rgrid.CurrentWorksheet.ResetAllPageBreaks();
} catch (Exception ex) {
    MessageBox.Show(ex.Message);
}

```

## Раздел "Правка" - функции, обычно доступные в меню "Правка"

Отмена последнего действия, возврат отменённого действия и повтор последнего. Выполняются только если это возможно

### VB

```
If rgrid.CanUndo = True Then rgrid.Undo()  
If rgrid.CanRedo = True Then rgrid.Redo()  
rgrid.RepeatLastAction(rgrid.CurrentWorksheet.SelectionRange)
```

#### C#

```
if (rgrid.CanUndo() == true) rgrid.Undo();  
if (rgrid.CanRedo() == true) rgrid.Redo();  
rgrid.RepeatLastAction(rgrid.CurrentWorksheet.SelectionRange);
```

## Буфер обмена (Clipboard) и "Выделить все"

Небольшое отступление. О реализации работы с буфером обмена.

При копировании содержимого нескольких ячеек, например, девяти, в буфер обмена заносится текст в следующем формате: ячейки одной строки разделяются символом табуляции, строки разделяются символом перевода строки:

```
A1 \t B1 \t C1 \n  
A2 \t B2 \t C2 \n  
A3 \t B3 \t C3 \n
```

**Примечание.** Для перехвата системных горячих клавиш введены события с префиксом **Before: BeforeCut, BeforeCopy, BeforePaste**. Также могут понадобиться события с префиксом **After: AfterCut, AfterCopy, AfterPaste**

## Для выделения всех ячеек используется метод SelectAll

#### VB

```
'Вырезать (Cut), Копировать (Copy). Проверяем возможность и выполняем процедуру  
If rgrid.CurrentWorksheet.CanCut = True Then rgrid.CurrentWorksheet.Cut()  
If rgrid.CurrentWorksheet.CanCopy = True Then rgrid.CurrentWorksheet.Copy()  
If rgrid.CurrentWorksheet.CanPaste = True Then rgrid.CurrentWorksheet.Paste()  
  
'Если нужно контролировать процесс получения данных из буфера обмена  
'Создание переменной и получение данных  
'Преобразование в массив типа Объект  
'Перенос данных в таблицу  
Dim text As String = Clipboard.GetText()  
Dim data(,) As Object = RGUtility.ParseTabbedString(text)  
rgrid.CurrentWorksheet.SetRangeData(rgrid.CurrentWorksheet.SelectionRange, data)  
  
'Выделить все  
rgrid.CurrentWorksheet.SelectAll()
```

#### C#

```
//Вырезать (Cut), Копировать (Copy). Проверяем возможность и выполняем процедуру  
if (rgrid.CurrentWorksheet.CanCut() == true) rgrid.CurrentWorksheet.Cut();  
if (rgrid.CurrentWorksheet.CanCopy() == true) rgrid.CurrentWorksheet.Copy();  
if (rgrid.CurrentWorksheet.CanPaste() == true) rgrid.CurrentWorksheet.Paste();  
  
//Если нужно контролировать процесс получения данных из буфера обмена  
//Создание переменной и получение данных  
//Преобразование в массив типа Объект  
//Перенос данных в таблицу  
string text = Clipboard.GetText();  
object[,] data = RGUtility.ParseTabbedString(text);  
rgrid.CurrentWorksheet.SetRangeData(rgrid.CurrentWorksheet.SelectionRange, data);  
  
//Выделить все  
rgrid.CurrentWorksheet.SelectAll();
```

Для событий **BeforeCut, BeforeCopy, BeforePaste** можно выполнить перехват горячих клавиш и не допустить выполнение функции. Для этого в обработчик события достаточно добавить строку `e.IsCancelled = True (VB), e.IsCancelled == true (C#)`.

## Раздел "Вид" - функции, обычно доступные в меню "Вид" (View)

1. Стиль выделения (**Selection Style**). Определяет внешний вид активной ячейки.
2. Режим выделения (**Selection Mode**). Определяет активный элемент (по умолчанию - диапазон).
3. Направление перехода (**Direction**). Определяет направление перехода фокуса после окончания правки ячейки.

#### VB

```

'Стиль выделения
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.None 'Отсутствует
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.Default 'По умолчанию
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.FocusRect 'Фокус Windows

'Режим выделения
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.None 'Отсутствует
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Cell 'Одна ячейка
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Range 'Диапазон ячеек
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Row 'Строка
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Column 'Столбец

'Направление перехода
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Right 'Вправо
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Down 'Вниз

```

## C#

```

//Стиль выделения
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.None;
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.Default;
rgrid.CurrentWorksheet.SelectionStyle = unvell.ReoGrid.WorksheetSelectionStyle.FocusRect;

//Режим выделения
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.None;
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Cell;
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Range;
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Row;
rgrid.CurrentWorksheet.SelectionMode = unvell.ReoGrid.WorksheetSelectionMode.Column;

//Направление перехода
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Right;
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Down;

```

## Фиксация ячеек (Freezing)

Если позиция имеет значения 0,0, то на данный момент зафиксированные строки или столбцы отсутствуют

Если зафиксированная область больше максимальной области отображения элемента управления, то активная область станет невидимой

## VB

```

'Получаем позицию заморозки/закрепления
Dim pos As CellPosition = rgrid.CurrentWorksheet.FreezePos()

'Проверяем - зафиксирован ли данный лист:
Dim isFrozen As Boolean = rgrid.CurrentWorksheet.IsFrozen()

'Фиксируем до указанной позиции (вводим третий аргумент, указывающий точку фиксации)
rgrid.CurrentWorksheet.FreezeToCell(5, 0) 'Обычная фиксация (указана только позиция)
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Left) 'С привязкой по левому краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Right) 'С привязкой по правому краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Top) 'С привязкой по верхнему краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Bottom) 'С привязкой по нижнему краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.LeftTop) 'С привязкой по левому и верхнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.LeftBottom) 'С привязкой по левому и нижнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.RightTop) 'С привязкой по правому и верхнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.RightBottom) 'С привязкой по правому и нижнему краям

'Отмена фиксации
rgrid.CurrentWorksheet.Unfreeze()

```

## C#

```
//Получаем позицию заморозки/закрепления
CellPosition pos = rgrid.CurrentWorksheet.FreezePos;

//Проверяем - зафиксирован ли данный лист:
bool isFrozen = rgrid.CurrentWorksheet.IsFrozen;

//Фиксируем до указанной позиции (вводим третий аргумент, указывающий точку фиксации)
rgrid.CurrentWorksheet.FreezeToCell(5, 0); //Обычная фиксация (указана только позиция)
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Left); //С привязкой по левому краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Right); //С привязкой по правому краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Top); //С привязкой по верхнему краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.Bottom); //С привязкой по нижнему краю
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.LeftTop); //С привязкой по левому и верхнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.LeftBottom); //С привязкой по левому и нижнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.RightTop); //С привязкой по правому и верхнему краям
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezeArea.RightBottom); //С привязкой по правому и нижнему краям

//Отмена фиксации
rgrid.CurrentWorksheet.Unfreeze();
```

## Изменение размеров таблицы (Resize)

На данный момент лист ReoGrid максимально может содержать 1048576 строк и 32768 столбцов

### VB

```
'Получаем количество строк и столбцов
Dim intRows As Integer = rgrid.CurrentWorksheet.RowCount 'Количество строк
Dim intCols As Integer = rgrid.CurrentWorksheet.ColumnCount 'Количество столбцов

'Задаём размер таблицы используя указанные выше переменные
rgrid.CurrentWorksheet.Resize(intRows, intCols)

'Или максимальное количество ячеек
rgrid.CurrentWorksheet.Resize(1048576, 32768)
```

### C#

```
//Получаем количество строк и столбцов
int intRows = rgrid.CurrentWorksheet.RowCount;
int intCols = rgrid.CurrentWorksheet.ColumnCount;

//Задаём размер таблицы используя указанные выше переменные
rgrid.CurrentWorksheet.Resize(intRows, intCols);

//Или максимальное количество ячеек
rgrid.CurrentWorksheet.Resize(1048576, 32768);
```

## Стили

Понятие "Стиль ячейки" подразумевает набор правил, который можно задать и изменить в процессе работы разрабатываемого приложения (программы). Сюда можно отнести основные параметры шрифтов, цвета текста и фона, выравнивание внутри ячейки. Полный список параметров приведён ниже.

### Компоненты стиля

Компонент	Значение PlainStyleFlag	Название свойства	Тип свойства
Цвет фона	BackColor	BackColor	Color или SolidColor
Шаблон стиля фона	FillPatternStyle	FillPatternStyle	HatchStyles
Цвет шаблона фона	FillPatternColor	FillPatternColor	Color или SolidColor
Цвет текста	TextColor	TextColor	Color или SolidColor
Название шрифта	FontName	FontName	string
Размер (кегель) шрифта	FontSize	FontSize	float (Single) для Windows Form; double для WPF
Стиль шрифта: полужирный	FontStyleBold	Bold	bool (Boolean)

Стиль шрифта: наклонный (курсив)	FontStyleItalic	Italic	bool (Boolean)
Стиль шрифта: подчёркнутый	FontStyleUnderline	Underline	bool (Boolean)
Стиль шрифта: зачёркнутый	FontStyleStrikethrough	Strikethrough	bool (Boolean)
Горизонтальное выравнивание	HorizontalAlign	HAlign	ReoGridHorAlign
Вертикальное выравнивание	VerticalAlign	VAlign	ReoGridVerAlign
Абзацный отступ	Indent	Indent	ushort
Перенос по словам	TextWrap	TextWrapMode	TextWrapMode
Угол поворота	RotateAngle	RotateAngle	int (Integer). Значения: -90°~90°
Отступы от краёв ячейки	Padding	Padding	PaddingValue

## Установка стиля

Существует 4 способа установки стиля:

- использование методов листа (**SetRangeStyles**);
- использование свойств экземпляра ячейки или диапазона;
- использование Действий;
- использование исполняемого сценария (ReoScript).

При использовании методов листа, необходимо:

- создать экземпляр объекта **WorksheetRangeStyle**;
- указать необходимые флаги **PlainStyleFlag**;
- указать необходимые значения стиля;
- применить стиль к диапазону ячеек.

Значение свойства **Flag** задаётся только 1 раз для каждого стиля. Для объединения флагов служит оператор "ИЛИ", написание которого зависит от конкретного языка программирования.

```
rgrid.CurrentWorksheet.SetRangeStyles(Диапазон типа RangePosition, Стиль типа RangePositionStyle)
```

## Шрифт

Можно выбрать его название (семейство) и размер (кегель).

## Выравнивание

Текст можно выравнивать по горизонтали и вертикали. Задаётся при помощи свойств **MyStyleFont.HAlign** и **MyStyleFont.VAlign**.

Если необходимо в одном экземпляре стиля задать как вертикальное, так и горизонтальное выравнивание текста, то будет полезен флаг **PlainStyleFlag.AlignAll**, аналогичный объединению флагов **PlainStyleFlag.HorizontalAlign** и **PlainStyleFlag.VerticalAlign**

## Заливка фона

Фон ячейки может быть как залит определённым цветом, так и текстурой (по шаблону). Ниже приведены примеры настройки границ и заливки фона

**Пример изменения всех основных параметров стиля, включая: параметры шрифта, выравнивание, цвета текста и фона**

VB



```

'Шаг 1. Создание экземпляра стиля
Dim MyStyle As New WorksheetRangeStyle

'Шаг 2. Задание нужных флагов
MyStyle.Flag = PlainStyleFlag.FontName Or PlainStyleFlag.FontSize Or
    PlainStyleFlag.TextColor Or
    PlainStyleFlag.FontStyleBold Or PlainStyleFlag.FontStyleItalic Or PlainStyleFlag.FontStyleUnderline Or PlainStyleFlag.FontStyle
    PlainStyleFlag.TextWrap Or
    PlainStyleFlag.HorizontalAlign Or PlainStyleFlag.VerticalAlign Or
    PlainStyleFlag.BackColor Or PlainStyleFlag.FillPattern

'Шаг 3. Изменение заданных флагами параметров
'Шрифты
MyStyle.FontName = cboFontName.Text
MyStyle.FontSize = Convert.ToInt16(Val(cboFontSize.Text))

MyStyle.FontSize = MyStyle.FontSize + 1 ' + 1 пт
cboFontSize.Text = Convert.ToString(MyStyle.FontSize)

MyStyle.RotateAngle = 45 'Поворот текста на 45 градусов

'Изменяем цвет шрифта
MyStyle.TextColor = Color.Red 'Красный

'Начертание шрифта
MyStyle.Bold = chkFontBold.Checked 'Полужирный
MyStyle.Italic = chkFontItalic.Checked 'Курсив (наклонный)
MyStyle.Underline = chkFontUnderline.Checked 'Подчёркнутый
MyStyle.Strikethrough = chkFontStrike.Checked 'Зачёркнутый

'Перенос текста (по умолчанию - без переноса, - no-wrap)
If chkTextWrap.Checked = True Then MyStyle.TextWrapMode = TextWrapMode.WordBreak Else MyStyle.TextWrapMode = TextWrapMode.NoWrap

'Горизонтальное выравнивание
MyStyle.HAlign = ReoGridHorAlign.Left 'По левому краю
MyStyle.HAlign = ReoGridHorAlign.Center 'По центру
MyStyle.HAlign = ReoGridHorAlign.Right 'По правому краю
MyStyle.HAlign = ReoGridHorAlign.DistributedIndent 'С распределённым отступом

'Вертикальное выравнивание
MyStyle.VAlign = ReoGridVerAlign.Top 'По верхней стороне
MyStyle.VAlign = ReoGridVerAlign.Middle 'Посередине
MyStyle.VAlign = ReoGridVerAlign.Bottom 'По нижней стороне

'Задание фона ячейки
MyStyle.BackColor = Color.Empty 'Очищаем ячейки
MyStyle.BackColor = unvell.ReoGrid.Graphics.SolidColor.Yellow 'Жёлтый цвет

'Фон на основе шаблона
MyStyle.FillPatternColor = Color.SkyBlue 'Цвет узора
MyStyle.FillPatternStyle = unvell.ReoGrid.Graphics.HatchStyles.DiagonalBrick 'Стиль узора

'Шаг 4. Применение стиля к выбранному диапазону
rgrid.CurrentWorksheet.SetRangeStyles(rgrid.CurrentWorksheet.SelectionRange, MyStyle)

```

C#

```

//Шаг 1. Создание экземпляра стиля
WorksheetRangeStyle MyStyle = new WorksheetRangeStyle();

//Шаг 2. Задание нужных флагов
MyStyle.Flag = PlainStyleFlag.FontName | PlainStyleFlag.FontSize |
PlainStyleFlag.TextColor |
PlainStyleFlag.FontStyleBold | PlainStyleFlag.FontStyleItalic | PlainStyleFlag.FontStyleUnderline | PlainStyleFlag.FontStyleStrike
PlainStyleFlag.TextWrap |
PlainStyleFlag.HorizontalAlign | PlainStyleFlag.VerticalAlign |
PlainStyleFlag.BackColor | PlainStyleFlag.FillPattern;

//Шаг 3. Изменение заданных флагами параметров
//Шрифты
MyStyle.FontName = cboFontName.Text;
MyStyle.FontSize = int.Parse(cboFontSize.Text);

MyStyle.FontSize = MyStyle.FontSize + 1; //+ 1 пт
cboFontSize.Text = Convert.ToString(MyStyle.FontSize);

MyStyle.RotateAngle = 45; //Поворот текста на 45 градусов

//Изменяем цвет шрифта
MyStyle.TextColor = Color.Red; //Красный

//Начертание шрифта
MyStyle.Bold = chkFontBold.Checked; //Полужирный
MyStyle.Italic = chkFontItalic.Checked; //Курсив (наклонный)
MyStyle.Underline = chkFontUnderline.Checked; //Подчёркнутый
MyStyle.Strikethrough = chkFontStrike.Checked; //Зачёркнутый

//Перенос текста (по умолчанию - без переноса, - no-wrap)
if (chkTextWrap.Checked == true) { MyStyle.TextWrapMode = TextWrapMode.WordBreak;} else { MyStyle.TextWrapMode = TextWrapMode.NoWr

//Горизонтальное выравнивание
MyStyle.HAlign = ReoGridHorAlign.Left; //По левому краю
MyStyle.HAlign = ReoGridHorAlign.Center; //По центру
MyStyle.HAlign = ReoGridHorAlign.Right; //По правому краю
MyStyle.HAlign = ReoGridHorAlign.DistributedIndent; //С распределённым отступом

//Вертикальное выравнивание
MyStyle.VAlign = ReoGridVerAlign.Top; //По верхней стороне
MyStyle.VAlign = ReoGridVerAlign.Middle; //Посередине
MyStyle.VAlign = ReoGridVerAlign.Bottom; //По нижней стороне

//Задание фона ячейки
MyStyle.BackColor = Color.Empty; //Очищаем ячейки
MyStyle.BackColor = unvell.ReoGrid.Graphics.SolidColor.Yellow; //Жёлтый цвет

//Фон на основе шаблона
MyStyle.FillPatternColor = Color.SkyBlue; //Цвет узора
MyStyle.FillPatternStyle = unvell.ReoGrid.Graphics.HatchStyles.DiagonalBrick; //Стиль узора

//Шаг 4. Применение стиля к выбранному диапазону
rgrid.CurrentWorksheet.SetRangeStyles(rgrid.CurrentWorksheet.SelectionRange, MyStyle);

```

## Примечания

- Помимо основных флагов, существуют и дополнительные, необходимые для уменьшения количества задаваемых флагов:
  - All** (все стили).
  - AlignAll** (горизонтальное и вертикальное выравнивание).
  - BackAll** (цвет и шаблон фона).
  - FontAll** (имя, размер и стиль-начертание).
  - FontStyleAll** (все стили-начертания шрифта).
  - LayoutAll** (перенос по словам, отступы, поворот текста).
  - LineAll** (цвет, стиль, толщина, наконечники). Пока не задействовано.
- Код подразумевает использование раскрывающихся списков `cboFontName` (имя шрифта), `cboFontSize` (размер шрифта), а также компоненты с двумя возможными состояниями (под `chk` понимается `CheckBox`). Для этого подходят и кнопки стандартного компонента Панелей инструментов.

## Поворот текста в ячейке

Для поворота текста в ячейках служит свойство **RotateAngle**. Оно может принимать значения от **-90 до 90**. Обычное направление соответствует значению **0**.

Если необходимо отобразить многострочный текст, то для этого следует установить для свойства **TextWrap** объекта стиля значение **WordBreak**.

Выравнивание текста задаётся всё теми же способами, что и для обычного текста.

**Совместимость с Excel:** настройки направления текста автоматически загружаются из файла Excel и выгружаются в него.

## VB

```
'Однострочный текст
rgrid.CurrentWorksheet.Cells("A1").Data = "Привет, мир! (90)"
rgrid.CurrentWorksheet.Cells("A1").Style.RotateAngle = 90

rgrid.CurrentWorksheet.Cells("B1").Data = "Привет, мир! (0)"
rgrid.CurrentWorksheet.Cells("B1").Style.RotateAngle = 0

rgrid.CurrentWorksheet.Cells("C1").Data = "Привет, мир! (-90)"
rgrid.CurrentWorksheet.Cells("C1").Style.RotateAngle = -90

rgrid.CurrentWorksheet.Cells("D1").Data = "Привет, мир! (45)"
rgrid.CurrentWorksheet.Cells("D1").Style.RotateAngle = 45

'Многострочный текст
rgrid.CurrentWorksheet.Cells("A2").Data = "Табличный компонент" & vbNewLine & "для .Net"
rgrid.CurrentWorksheet.Cells("A2").Style.TextWrap = TextWrapMode.WordBreak
rgrid.CurrentWorksheet.Cells("A2").Style.RotateAngle = -45

'Выравнивание
rgrid.CurrentWorksheet.Cells("A3").Data = "Привет, мир!"
rgrid.CurrentWorksheet.Cells("A3").Style.HAlign = ReoGridHorAlign.Left
rgrid.CurrentWorksheet.Cells("A3").Style.RotateAngle = 90

rgrid.CurrentWorksheet.Cells("B3").Data = "Привет, мир!"
rgrid.CurrentWorksheet.Cells("B3").Style.HAlign = ReoGridHorAlign.Center
rgrid.CurrentWorksheet.Cells("B3").Style.RotateAngle = 90

rgrid.CurrentWorksheet.Cells("C3").Data = "Привет, мир!"
rgrid.CurrentWorksheet.Cells("C3").Style.HAlign = ReoGridHorAlign.Right
rgrid.CurrentWorksheet.Cells("C3").Style.RotateAngle = 90
```

## C#

```
rgrid.CurrentWorksheet.Cells["A1"].Data = "Привет, мир! (90)";
rgrid.CurrentWorksheet.Cells["A1"].Style.RotateAngle = 90;

rgrid.CurrentWorksheet.Cells["B1"].Data = "Привет, мир! (0)";
rgrid.CurrentWorksheet.Cells["B1"].Style.RotateAngle = 0;

rgrid.CurrentWorksheet.Cells["C1"].Data = "Привет, мир! (-90)";
rgrid.CurrentWorksheet.Cells["C1"].Style.RotateAngle = -90;

rgrid.CurrentWorksheet.Cells["D1"].Data = "Привет, мир! (45)";
rgrid.CurrentWorksheet.Cells["D1"].Style.RotateAngle = 45;

//Многострочный текст
rgrid.CurrentWorksheet.Cells["A2"].Data = "Табличный компонент\nдля .Net";
rgrid.CurrentWorksheet.Cells["A2"].Style.TextWrap = TextWrapMode.WordBreak;
rgrid.CurrentWorksheet.Cells["A2"].Style.RotateAngle = -45;

//Выравнивание
rgrid.CurrentWorksheet.Cells["A3"].Data = "Привет, мир!";
rgrid.CurrentWorksheet.Cells["A3"].Style.HAlign = ReoGridHorAlign.Left;
rgrid.CurrentWorksheet.Cells["A3"].Style.RotateAngle = 90;

rgrid.CurrentWorksheet.Cells["B3"].Data = "Привет, мир!";
rgrid.CurrentWorksheet.Cells["B3"].Style.HAlign = ReoGridHorAlign.Center;
rgrid.CurrentWorksheet.Cells["B3"].Style.RotateAngle = 90;

rgrid.CurrentWorksheet.Cells["C3"].Data = "Привет, мир!";
rgrid.CurrentWorksheet.Cells["C3"].Style.HAlign = ReoGridHorAlign.Right;
rgrid.CurrentWorksheet.Cells["C3"].Style.RotateAngle = 90;
```

	A	B	C	D
1	Привет, мир! (90)		Привет, мир! (-) (0)	Привет, мир! (45)
2	Табличный компонент Для .Net			
3	Привет, мир!	Привет, мир!	Привет, мир!	

Результат выполнения кода

	A	B		A	B		A	B
1	Привет, мир! (90)		1	Привет, мир! (90)		1	Привет, мир! (90)	
2	ReoGrid		2	Excel 2010		2	LibreOffice Calc 5	

Отображение созданного в ReoGrid файла (первые строки из кода примера). Выравнивание текста не задавалось

Видеодемонстрация ([https://www.youtube.com/watch?v=k6\\_U20bAR58](https://www.youtube.com/watch?v=k6_U20bAR58))

## Изменение стиля отдельной ячейки

Для изменения стиля отдельной ячейки можно использовать её свойство **Style**.

### VB

```

rgrid.CurrentWorksheet.Cells("E5").Style.BackColor = Color.LightYellow 'Цвет фона
rgrid.CurrentWorksheet.Cells("E5").Style.Bold = True 'Полужирный
rgrid.CurrentWorksheet.Cells("E5").Style.Italic = True 'Наклонный
rgrid.CurrentWorksheet.Cells("E5").Style.Underline = True 'Подчёркнутый
rgrid.CurrentWorksheet.Cells("E5").Style.Strikethrough = True 'Зачёркнутый
rgrid.CurrentWorksheet.Cells("E5").Style.FontName = "Verdana" 'Гарнитура
rgrid.CurrentWorksheet.Cells("E5").Style.FontSize = 12 'Кегль
rgrid.CurrentWorksheet.Cells("E5").Style.HAlign = ReoGridHorAlign.Center 'Горизонтальное выравнивание
rgrid.CurrentWorksheet.Cells("E5").Style.VAlign = ReoGridVerAlign.Middle 'Вертикальное выравнивание
rgrid.CurrentWorksheet.Cells("E5").Style.Indent = 15 'Отступ текста
rgrid.CurrentWorksheet.Cells("E5").Style.Padding = New PaddingValue(5, 5, 5, 5) 'Отступы в ячейке
rgrid.CurrentWorksheet.Cells("E5").Style.Padding = New PaddingValue(5) 'Отступы в ячейке
rgrid.CurrentWorksheet.Cells("E5").Style.RotateAngle = 90 'Поворот текста (в Pro-издании)
rgrid.CurrentWorksheet.Cells("E5").Style.TextColor = Color.Blue 'Цвет текста
rgrid.CurrentWorksheet.Cells("E5").Style.TextWrap = TextWrapMode.WordBreak 'Перенос по словам

```

### C#

```

//Основные настройки стиля для отдельно взятой ячейки. Для большего числа ячеек нужно использовать стили
rgrid.CurrentWorksheet.Cells["E5"].Style.BackColor = Color.LightYellow; //Цвет фона
rgrid.CurrentWorksheet.Cells["E5"].Style.Bold = true; //Полужирный
rgrid.CurrentWorksheet.Cells["E5"].Style.Italic = true; //Наклонный
rgrid.CurrentWorksheet.Cells["E5"].Style.Underline = true; //Подчёркнутый
rgrid.CurrentWorksheet.Cells["E5"].Style.Strikethrough = true; //Зачёркнутый
rgrid.CurrentWorksheet.Cells["E5"].Style.FontName = "Verdana"; //Гарнитура
rgrid.CurrentWorksheet.Cells["E5"].Style.FontSize = 12; //Кегль
rgrid.CurrentWorksheet.Cells["E5"].Style.HAlign = ReoGridHorAlign.Center; //Горизонтальное выравнивание
rgrid.CurrentWorksheet.Cells["E5"].Style.VAlign = ReoGridVerAlign.Middle; //Вертикальное выравнивание
rgrid.CurrentWorksheet.Cells["E5"].Style.Indent = 15; //Отступ текста
rgrid.CurrentWorksheet.Cells["E5"].Style.Padding = new PaddingValue(5, 5, 5, 5); //Отступы в ячейке
rgrid.CurrentWorksheet.Cells["E5"].Style.Padding = new PaddingValue(5); //Отступы в ячейке
rgrid.CurrentWorksheet.Cells["E5"].Style.RotateAngle = 90; //Поворот текста (в Pro-издании)
rgrid.CurrentWorksheet.Cells["E5"].Style.TextColor = Color.Blue; //Цвет текста
rgrid.CurrentWorksheet.Cells["E5"].Style.TextWrap = TextWrapMode.WordBreak; //Перенос по словам

```

## Использование Действий (DoAction)

Пример использования этого механизма:

```
rgrid.CurrentWorksheet.DoAction(new RGSetRangeStyleAction(Диапазон типа RangePosition, Стил ь типа WorksheetRangeStyle))
```

## Использование сценариев

Так же может быть использована команда на сценарном языке ReoScript.

```
rgrid.CurrentWorksheet.getCell(0, 0).style.backgroundColor = 'red'
```

## Получение значений стилей

Получаем стили диапазона или ячейки:

```
rgrid.CurrentWorksheet.GetRangeStyles(Диапазон типа RangePosition)  
rgrid.CurrentWorksheet.GetCellStyles(Позиция_ячейки типа CellPosition)
```

Приведённый ниже код необходим для настройки пользовательского интерфейса при выборе пользователем другой ячейки или диапазона. Данный код работает в обработчике события SelectionRangeChanged компонента **ReoGrid**

**VB**

```
If e.Range.IsEmpty = True Then MsgBox("Выбран пустой диапазон") : Exit Sub 'Ничего не выбрано - можно не продолжать  
lblStatus.Text = "Выделенный диапазон: " & rgrid.CurrentWorksheet.SelectionRange.ToString()
```

```
Dim MyStyle As New WorksheetRangeStyle 'Создаём экземпляр стиля  
MyStyle = rgrid.CurrentWorksheet.GetRangeStyles(rgrid.CurrentWorksheet.SelectionRange)
```

```
cboFontName.Text = MyStyle.FontName 'Гарнитура шрифта  
cboFontSize.Text = Convert.ToString(MyStyle.FontSize) 'Кегль шрифта (размер)  
chkFontBold.Checked = MyStyle.Bold 'Полужирный  
chkFontItalic.Checked = MyStyle.Italic 'Курсив  
chkFontUnderline.Checked = MyStyle.Underline 'Подчёркнутый  
chkFontStrike.Checked = MyStyle.Strikethrough 'Зачёркнутый
```

```
'Выравнивание по левому краю  
If MyStyle.HAlign = ReoGridHorAlign.Left Then  
    radTextAlignLeft.Checked = True  
Else  
    radTextAlignLeft.Checked = False  
End If
```

```
'Горизонтальная центровка  
If MyStyle.HAlign = ReoGridHorAlign.Center Then  
    radTextAlignCenter.Checked = True  
Else  
    radTextAlignCenter.Checked = False  
End If
```

```
'Выравнивание по правому краю  
If MyStyle.HAlign = ReoGridHorAlign.Right Then  
    radTextAlignRight.Checked = True  
Else  
    radTextAlignRight.Checked = False  
End If
```

```
'Распределённый отступ  
If MyStyle.HAlign = ReoGridHorAlign.DistributedIndent Then  
    radTextDisributedIndent.Checked = True  
Else  
    radTextDisributedIndent.Checked = False  
End If
```

```
'Перенос по словам  
chkTextWrap.Checked = Convert.ToBoolean(MyStyle.TextWrapMode)
```

```
'Вертикальное выравнивание: Сверху  
If MyStyle.VAlign = ReoGridVerAlign.Top Then  
    radTextAlignTop.Checked = True  
Else  
    radTextAlignTop.Checked = False  
End If
```

```
'Вертикальное выравнивание: Посередине  
If MyStyle.VAlign = ReoGridVerAlign.Middle Then  
    radTextAlignMiddle.Checked = True  
Else  
    radTextAlignMiddle.Checked = False  
End If
```

```
'Вертикальное выравнивание: Снизу  
If MyStyle.VAlign = ReoGridVerAlign.Bottom Then  
    radTextAlignBottom.Checked = True  
Else  
    radTextAlignBottom.Checked = False  
End If
```

```
'Другие свойства:  
'grpBackgroundColor.BackColor = MyStyle.BackColor 'Цвет фона  
'MyStyle.CopyFrom() 'Копировать из другого стиля  
'MyStyle.FillPatternColor 'Цвет шаблона фона (узора)  
'MyStyle.FillPatternStyle 'Шаблон (узор)  
'MyStyle.HAlign.General  
'MyStyle.Padding 'Отступы  
'grpTextColor.BackColor = MyStyle.TextColor 'Цвет текста
```

```

if (e.Range.IsEmpty == true)
{
    MessageBox.Show("Выбран пустой диапазон"); return;
}
//Ничего не выбрано - можно не продолжать
lblStatus.Text = "Выделенный диапазон: " + rgrid.CurrentWorksheet.SelectionRange.ToString();

WorksheetRangeStyle MyStyle = new WorksheetRangeStyle();
//Создаём экземпляр стиля
MyStyle = rgrid.CurrentWorksheet.GetRangeStyles(rgrid.CurrentWorksheet.SelectionRange);

cboFontName.Text = MyStyle.FontName; //Гарнитура шрифта
cboFontSize.Text = Convert.ToString(MyStyle.FontSize); //Кегль шрифта (размер)
chkFontBold.Checked = MyStyle.Bold; //Полужирный
chkFontItalic.Checked = MyStyle.Italic; //Курсив
chkFontUnderline.Checked = MyStyle.Underline; //Подчёркнутый
chkFontStrike.Checked = MyStyle.Strikethrough; //Зачёркнутый

//Выравнивание по левому краю
if (MyStyle.HAlign == ReoGridHorAlign.Left)
{radTextAlignLeft.Checked = true;}
else
{radTextAlignLeft.Checked = false;}

//Горизонтальная центровка
if (MyStyle.HAlign == ReoGridHorAlign.Center)
{radTextAlignCenter.Checked = true;}
else
{radTextAlignCenter.Checked = false;}

//Выравнивание по правому краю
if (MyStyle.HAlign == ReoGridHorAlign.Right)
{radTextAlignRight.Checked = true;}
else
{radTextAlignRight.Checked = false;}

//Распределённый отступ
if (MyStyle.HAlign == ReoGridHorAlign.DistributedIndent)
{radTextDisributedIndent.Checked = true;}
else
{radTextDisributedIndent.Checked = false;}

//Перенос по словам
chkTextWrap.Checked = Convert.ToBoolean(MyStyle.TextWrapMode);

//Вертикальное выравнивание: Сверху
if (MyStyle.VAlign == ReoGridVerAlign.Top)
{radTextAlignTop.Checked = true;}
else
{radTextAlignTop.Checked = false;}

//Вертикальное выравнивание: Посередине
if (MyStyle.VAlign == ReoGridVerAlign.Middle)
{radTextAlignMiddle.Checked = true;}
else
{radTextAlignMiddle.Checked = false;}

//Вертикальное выравнивание: Снизу
if (MyStyle.VAlign == ReoGridVerAlign.Bottom)
{radTextAlignBottom.Checked = true;}
else
{radTextAlignBottom.Checked = false;}

//Другие свойства:
//grpBackgroundColor.BackColor = MyStyle.BackColor; //Цвет фона
//MyStyle.CopyFrom(); //Копировать из другого стиля
//MyStyle.FillPatternColor; //Цвет шаблона фона (узора)
//MyStyle.FillPatternStyle; //Шаблон (узор)
//MyStyle.HAlign.General
//MyStyle.Padding; //Отступы
//grpTextColor.BackColor = MyStyle.TextColor; //Цвет текста

```

## Удаление стиля

Для удаления стилей используется свойство листа **RemoveRangeStyles**.

Флаг **PlainStyleFlag** позволяет указать, какие стили должны быть удалены. В примере показано удаление стиля фона для выбранного диапазона ячеек с помощью метода и действия.

#### VB

```
'Удаляем стиль для выделенного диапазона, даже если он состоит из одной ячейки.  
rgrid.CurrentWorksheet.RemoveRangeStyles(rgrid.CurrentWorksheet.SelectionRange, PlainStyleFlag.BackAll) 'Метод  
  
'Убираем фоновый цвет для выбранного диапазона:  
rgrid.DoAction(New RemoveRangeStyleAction(rgrid.CurrentWorksheet.SelectionRange, PlainStyleFlag.BackAll)) 'Действие
```

#### C#

```
//Удаляем стиль для выделенного диапазона, даже если он состоит из одной ячейки.  
rgrid.CurrentWorksheet.RemoveRangeStyles(rgrid.CurrentWorksheet.SelectionRange, PlainStyleFlag.BackAll); //Метод  
  
//Убираем фоновый цвет для выбранного диапазона:  
rgrid.DoAction(new RemoveRangeStyleAction(rgrid.CurrentWorksheet.SelectionRange, PlainStyleFlag.BackAll)); //Действие
```

## Границы (Borders)

Для настройки границ применяются стили, которые создаются аналогично стилям текста, но задавать флаги не нужно. Что необходимо сделать:

1. Создать экземпляр стиля
2. Указать тип линий
3. Указать цвет линий
4. Применить стиль к указанному диапазону ячеек, дополнительно задав затрагиваемые границы.

Тип линии задаётся перечислением **BorderLineStyle** и может принимать значения, показанные в приведённом ниже примере.

Цвет линии задаётся перечислением **Color** или другим предусмотренным в .Net способом.

Тип границы задаётся перечислением **BorderPositions** и может принимать значения, показанные в приведённом ниже примере.

В примере показаны все основные варианты типов линий, вариантов задания цвета и затрагиваемых границ.

#### VB



```
'Объявление экземпляра стиля диапазона ячеек
```

```
Dim MyStyleBorder1 As New RangeBorderStyle
```

```
'Тип линий, оставляем нужную
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldDashDot 'Полужирный штрих-точка
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldDashDotDot 'Полужирный штрих-точка-точка
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldDashed 'Полужирный штрих
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldDotted 'Полужирная точка
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldSolid 'Полужирная сплошная
```

```
MyStyleBorder1.Style = BorderLineStyle.BoldSolidStrong 'Полужирная сплошная
```

```
MyStyleBorder1.Style = BorderLineStyle.DashDot 'Штрих-точка
```

```
MyStyleBorder1.Style = BorderLineStyle.DashDotDot 'Штрих-точка-точка
```

```
MyStyleBorder1.Style = BorderLineStyle.Dashed 'Штриховая
```

```
MyStyleBorder1.Style = BorderLineStyle.Dashed2 'Штриховая
```

```
MyStyleBorder1.Style = BorderLineStyle.Dotted 'Точечная
```

```
MyStyleBorder1.Style = BorderLineStyle.DoubleLine 'Двойная линия
```

```
MyStyleBorder1.Style = BorderLineStyle.None 'Без границы
```

```
MyStyleBorder1.Style = BorderLineStyle.Solid 'Сплошная
```

```
'Цвета линий
```

```
MyStyleBorder1.Color = Color.Green 'Цвет из палитры .Net
```

```
MyStyleBorder1.Color = unvell.ReoGrid.Graphics.SolidColor.Green 'Цвет из палитры Unvell.Reogrid
```

```
MyStyleBorder1.Color = Color.FromArgb(127, 127, 127, 127) 'Произвольный цвет с прозрачностью
```

```
'Параметры границ диапазона или ячейки
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All, MyStyleBorder1) 'Все границы
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Backslash, MyStyleBorder1) 'Диагональ
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Bottom, MyStyleBorder1) 'Нижняя
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideAll, MyStyleBorder1) 'Все внут
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideHorizontal, MyStyleBorder1) 'В
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideVertical, MyStyleBorder1) 'Вну
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Left, MyStyleBorder1) 'Левая
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.LeftRight, MyStyleBorder1) 'Левая и
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.None, MyStyleBorder1) 'Отсутствуют
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Outside, MyStyleBorder1) 'Внешние (р
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Right, MyStyleBorder1) 'Правая
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Slash, MyStyleBorder1) 'Диагональная
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Top, MyStyleBorder1) 'Верхняя
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.TopBottom, MyStyleBorder1) 'Верхняя
```

```
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.X, MyStyleBorder1) 'Все диагональные
```

C#

```

//Объявление экземпляра стиля диапазона ячеек
RangeBorderStyle MyStyleBorder1 = new RangeBorderStyle();

//Тип линий, оставляем нужную
MyStyleBorder1.Style = BorderLineStyle.BoldDashDot; //Полужирный штрих-точка
MyStyleBorder1.Style = BorderLineStyle.BoldDashDotDot; //Полужирный штрих-точка-точка
MyStyleBorder1.Style = BorderLineStyle.BoldDashed; //Полужирный штрих
MyStyleBorder1.Style = BorderLineStyle.BoldDotted; //Полужирная точка
MyStyleBorder1.Style = BorderLineStyle.BoldSolid; //Полужирная сплошная
MyStyleBorder1.Style = BorderLineStyle.BoldSolidStrong; //Полужирная сплошная
MyStyleBorder1.Style = BorderLineStyle.DashDot; //Штрих-точка
MyStyleBorder1.Style = BorderLineStyle.DashDotDot; //Штрих-точка-точка
MyStyleBorder1.Style = BorderLineStyle.Dashed; //Штриховая
MyStyleBorder1.Style = BorderLineStyle.Dashed2; //Штриховая
MyStyleBorder1.Style = BorderLineStyle.Dotted; //Точечная
MyStyleBorder1.Style = BorderLineStyle.DoubleLine; //Двойная линия
MyStyleBorder1.Style = BorderLineStyle.None; //Без границы
MyStyleBorder1.Style = BorderLineStyle.Solid; //Сплошная

//Цвета линий
MyStyleBorder1.Color = Color.Green; //Цвет из палитры .Net
MyStyleBorder1.Color = unvell.ReoGrid.Graphics.SolidColor.Green; //Цвет из палитры Unvell.ReoGrid
MyStyleBorder1.Color = Color.FromArgb(127, 127, 127, 127); //Произвольный цвет с прозрачностью

//Параметры границ диапазона или ячейки
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All, MyStyleBorder1); //Все границы
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Backslash, MyStyleBorder1); //Диагон
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Bottom, MyStyleBorder1); //Нижняя
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Outside, MyStyleBorder1); //Все вн
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideAll, MyStyleBorder1); //Все вн
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideHorizontal, MyStyleBorder1); /
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.InsideVertical, MyStyleBorder1); //В
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Left, MyStyleBorder1); //Левая
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.LeftRight, MyStyleBorder1); //Левая
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.None, MyStyleBorder1); //Отсутствуют
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Outside, MyStyleBorder1); //Внешние
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Right, MyStyleBorder1); //Правая
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Slash, MyStyleBorder1); //Диагональн
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.Top, MyStyleBorder1); //Верхняя
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.TopBottom, MyStyleBorder1); //Верхняя
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.X, MyStyleBorder1); //Все диагональн

```

## Очистка границ (Clear)

Для очистки границ достаточно задать для них пустой цвет (*Color.Empty*) или применить функцию

**VB**

```

Dim MyStyleBorder2 As New RangeBorderStyle
MyStyleBorder2.Color = Color.Empty
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All, MyStyleBorder2)
'или
rgrid.CurrentWorksheet.RemoveRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All)

```

**C#**

```

RangeBorderStyle MyStyleBorder2 = new RangeBorderStyle();
MyStyleBorder2.Color = Color.Empty;
rgrid.CurrentWorksheet.SetRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All, MyStyleBorder2);
//или
rgrid.CurrentWorksheet.RemoveRangeBorders(rgrid.CurrentWorksheet.SelectionRange, BorderPositions.All);

```

Видеодемонстрация (<https://www.youtube.com/watch?v=OfBscQnN32o>)

## Объединение, разбивка и группировка ячеек

ReoGrid позволяет объединять и снова разделять ячейки. При этом объединять пересекающиеся диапазоны запрещено: ячейки из одного объединения не должны принадлежать другому. Также предусмотрены возможности проверки текущего состояния произвольной ячейки.

### Объединение ячеек

1. Объединяем ячейки, начиная с (2,1). Количество строк - 3, количество столбцов - 4
2. Диапазон объединения начинается в (0,0), содержит 4 строки и 6 столбцов

## VB

```
'1
rgrid.CurrentWorksheet.MergeRange(New RangePosition(2, 1, 3, 4))
'2
'rgrid.CurrentWorksheet.DoAction(New MergeRangeAction(New RangePosition(0, 0, 4, 6)))
```

## C#

```
//1
rgrid.CurrentWorksheet.MergeRange(new RangePosition(2, 1, 3, 4));
//2
//rgrid.CurrentWorksheet.DoAction(new MergeRangeAction(new RangePosition(0, 0, 4, 6)))
```

## Исключение **RangeIntersectionException**

Если попытаться объединить 2 и более ячеек, каждая из которых уже входит в состав своего объединения, то возникнет исключение **RangeIntersectionException**.

Пользовательский код позволяет отловить это исключение и прервать текущую операцию.

Метод **CheckIntersectedMergingRange** применяется для проверки принадлежности любой ячейки определённому диапазону. Этот метод помогает избежать указанное выше исключение.

## VB

```
Dim outputRange As RangePosition = rgrid.CurrentWorksheet.CheckIntersectedMergingRange(rgrid.CurrentWorksheet.SelectionRange)
```

## C#

```
RangePosition outputRange = rgrid.CurrentWorksheet.CheckIntersectedMergingRange(rgrid.CurrentWorksheet.SelectionRange);
```

Если выражение `outputRange.IsEmpty` является истинным, то диапазон `inputRange` безопасен. Существует ещё один метод, делающий то же самое.

## VB

```
Dim rs As Boolean = rgrid.CurrentWorksheet.HasIntersectedMergingRange(rgrid.CurrentWorksheet.SelectionRange)
MsgBox("Слияние ячеек")
```

## C#

```
bool rs = rgrid.CurrentWorksheet.HasIntersectedMergingRange(rgrid.CurrentWorksheet.SelectionRange);
MessageBox.Show("Слияние ячеек");
```

Нельзя объединять ячейки из разных объединённых диапазонов - это вызовет ошибку.

## VB

```
rgrid.CurrentWorksheet.Reset()
Try
    'Попытка объединить пересекающиеся диапазоны приведёт к возникновению исключения
    rgrid.CurrentWorksheet.MergeRange(New RangePosition(2, 2, 5, 5))
    rgrid.CurrentWorksheet.MergeRange(New RangePosition(3, 3, 5, 5))
Catch 'unvell.reogrid.RangeIntersectionException
    MsgBox.Show("Обнаружено исключение RangeIntersectionException! Невозможно изменить часть диапазона ячеек.")
End Try
```

## C#

```
rgrid.CurrentWorksheet.Reset();
try
{
    //Попытка объединить пересекающиеся диапазоны приведёт к возникновению исключения
    rgrid.CurrentWorksheet.MergeRange(new RangePosition(2, 2, 5, 5));
    rgrid.CurrentWorksheet.MergeRange(new RangePosition(3, 3, 5, 5));
    //unvell.reogrid.RangeIntersectionException
}
catch
{
    MessageBox.Show("Обнаружено исключение RangeIntersectionException! Невозможно изменить часть диапазона ячеек.");
}
```

## Проверка и получение значений

Используются следующие методы и свойства:

- **IsMergedCell** - метод для проверки факта объединения ячеек;

- **GetRowspan** - свойство ячейки для определения количества строк в объединении (number of rowspan);
- **GetColspan** - свойство ячейки для определения количества столбцов в объединении (number of colspan);
- **GetMergedCellOfRange** - метод для поиска объединенной ячейки в определенном диапазоне.
- **CheckMergedRange** - метод для определения всей ячейки, если известна её часть.
- **IsValidCell** - метод проверки ячеек на действительность. Недействительными ячейками являются ячейки, объединённые с другими. Недействительной ячейке нельзя задать данные или стили.

Примеры использования.

Ячейка в позиции 2:1 является объединённой, она содержит 4 строки (rowspan=4) и 3 столбца (colspan=3).

#### VB

```
Dim isMergedCell As Boolean = rgrid.CurrentWorksheet.IsMergedCell(2, 1)
Dim isMergedCell_1 As Boolean = rgrid.CurrentWorksheet.Cells(1, 2).IsMergedCell
Dim rowspan As Integer = rgrid.CurrentWorksheet.Cells(1, 2).GetRowspan()
Dim colspan As Integer = rgrid.CurrentWorksheet.Cells(1, 2).GetColspan()
'Dim output As Cell = rgrid.CurrentWorksheet.GetMergedCellOfRange(Input)
'Dim outputRange As RangePosition = rgrid.CurrentWorksheet.CheckMergedRange(inputRange)
rgrid.CurrentWorksheet.IsValidCell(2, 2) ' true
rgrid.CurrentWorksheet.IsValidCell(2, 3) ' false
```

#### C#

```
bool isMergedCell = rgrid.CurrentWorksheet.IsMergedCell(2, 1);
bool isMergedCell_1 = rgrid.CurrentWorksheet.Cells[1, 2].IsMergedCell;
int rowspan = rgrid.CurrentWorksheet.Cells[1, 2].GetRowspan();
int colspan = rgrid.CurrentWorksheet.Cells[1, 2].GetColspan();
//Cell output = rgrid.CurrentWorksheet.GetMergedCellOfRange(Input());
//RangePosition outputRange = rgrid.CurrentWorksheet.CheckMergedRange(inputRange);
rgrid.CurrentWorksheet.IsValidCell(2, 2); //true
rgrid.CurrentWorksheet.IsValidCell(2, 3); //false
```

## Проверка количества выбранных ячеек

Так как **ReoGrid** обычно работает с выделенными диапазонами ячеек, то нужно предусматривать проверку: выбрана одна ячейка или более одной.

#### VB

```
If rgrid.CurrentWorksheet.SelectionRange.Rows = 1 And rgrid.CurrentWorksheet.SelectionRange.Cols = 1 Then
    MessageBox.Show("Выбранный диапазон должен состоять хотя бы из двух ячеек.")
Else
    Try
        rgrid.CurrentWorksheet.MergeRange(rgrid.CurrentWorksheet.SelectionRange)
    Catch 'RangeIntersectionException
        MessageBox.Show("Невозможно объединить пересекающиеся диапазоны.")
    End Try
End If
```

#### C#

```
if (rgrid.CurrentWorksheet.SelectionRange.Rows == 1 & rgrid.CurrentWorksheet.SelectionRange.Cols == 1)
{
    MessageBox.Show("Выбранный диапазон должен состоять хотя бы из двух ячеек.");
}
else
{
    try
    {
        rgrid.CurrentWorksheet.MergeRange(rgrid.CurrentWorksheet.SelectionRange);
        //RangeIntersectionException
    }
    catch
    {
        MessageBox.Show("Невозможно объединить пересекающиеся диапазоны.");
    }
}
```

## Разбивка (разделение) ячеек

Разделение объединённых ячеек.

#### VB