

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

## Необходимые пространства имён (Namespaces)

Для создания простого редактора будет достаточно подключить следующие пространства имён:

```
'VB
Imports System.IO 'Для работы с файловой системой
Imports System.Text 'Для работы с кодировками текста
Imports unvell.ReoGrid 'Сам компонент
Imports unvell.ReoGrid.CellTypes 'Типы данных ячеек (кнопка, флажок)
Imports unvell.ReoGrid.Actions 'Действия
Imports unvell.ReoGrid.DataFormat 'Форматы данных (дата, проценты, текст)
'Imports unvell.ReoGrid.Events 'События
'Imports unvell.ReoGrid.Editor 'Если понадобится готовый редактор

//C#
using System.IO 'Для работы с файловой системой
using System.Text 'Для работы с кодировками текста
using unvell.ReoGrid 'Сам компонент
using unvell.ReoGrid.CellTypes 'Типы данных ячеек (кнопка, флажок)
using unvell.ReoGrid.Actions 'Действия
using unvell.ReoGrid.DataFormat 'Форматы данных (дата, проценты, текст)
//using unvell.ReoGrid.Events 'События
//using unvell.ReoGrid.Editor 'Если понадобится готовый редактор
```

## Инициализация списков

Этот фрагмент кода можно включить в состав процедуры Form\_Load

Все компоненты с префиксом cbo являются раскрывающимися списками **ComboBox**

### Перечень кеглей (размеров) шрифта

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

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

### Получение списка гарнитур (семейств) шрифтов

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

//C#
cboFontName.DataSource = FontFamily.Families;
cboFontName.ValueMember = "Name";
cboFontName.Text = "Выберите гарнитуру";
```

### Масштаб

```
'VB
cboZoom.Items.AddRange({"25%", "50%", "75%", "100%", "125%", "150%", "200%"})

//C#
cboZoom.Items.AddRange({"25%", "50%", "75%", "100%", "125%", "150%", "200%"});
```

## Функции для строки формул

```
'VB
cboFunctions.Items.Add("INDIRECT()") 'Возвращает ссылку, заданную текстовой строкой
cboFunctions.Items.Add("ADDRESS(,)") 'Возвращает ядрес ячейки на листе
cboFunctions.Items.Add("COUNT(,)") 'Возвращает количество ячеек, содержащих числа, и подсчитывает числа в списке аргументов
cboFunctions.Items.Add("COUNTIF(,)") 'Возвращает количество ячеек в заданном диапазоне, которые отвечают единому заданному критерию
cboFunctions.Items.Add("SUM(,)") 'Возвращает сумму значений чисел. Пример: SUM(B11:E12)
cboFunctions.Items.Add("SUMIF(, ,)") 'Возвращает сумму значений в диапазоне, которые соответствуют указанным критериям
cboFunctions.Items.Add("AVERAGE(,)") 'Возвращает среднее арифметическое значение указанных аргументов
cboFunctions.Items.Add("LOOKUP(, ,)") 'Возвращает значение либо из одной строки, либо одного столбца диапазона или массива
cboFunctions.Items.Add("ROWS(,)") 'Возвращает количество строк в ссылке (reference) или массиве
cboFunctions.Items.Add("COLUMNS(,)") 'Возвращает количество столбцов в ссылке (reference) или массиве
cboFunctions.Items.Add("INDEX(, ,)") 'Возвращает значение или ссылку (reference) на значение из таблицы или диапазона ячеек
cboFunctions.Items.Add("CEILING(, ,)") 'Возвращает число, округляемое в сторону увеличения, от нуля до ближайшего кратного указанному значению
cboFunctions.Items.Add("LEN(,)") 'Возвращает количество символов в текстовой строке
cboFunctions.Items.Add("LENB(,)") 'Возвращает количество байт, используемых для представления знаков в текстовой строке
cboFunctions.Items.Add("ROUND(, ,)") 'Округление числа до ближайшего значения. Примеры: ROUND(B8), ROUND(E8, 2)

//C#
cboFunctions.Items.Add("INDIRECT()"); //Возвращает ссылку, заданную текстовой строкой
cboFunctions.Items.Add("ADDRESS(, ,)"); //Возвращает ядрес ячейки на листе
cboFunctions.Items.Add("COUNT(, ,)"); //Возвращает количество ячеек, содержащих числа, и подсчитывает числа в списке аргументов
cboFunctions.Items.Add("COUNTIF(, ,)"); //Возвращает количество ячеек в заданном диапазоне, которые отвечают единому заданному критерию
cboFunctions.Items.Add("SUM(, ,)"); //Возвращает сумму значений чисел. Пример: SUM(B11:E12)
cboFunctions.Items.Add("SUMIF(, , ,)"); //Возвращает сумму значений в диапазоне, которые соответствуют указанным критериям
cboFunctions.Items.Add("AVERAGE(, ,)"); //Возвращает среднее арифметическое значение указанных аргументов
cboFunctions.Items.Add("LOOKUP(, , ,)"); //Возвращает значение либо из одной строки, либо одного столбца диапазона или массива
cboFunctions.Items.Add("ROWS(, ,)"); //Возвращает количество строк в ссылке (reference) или массиве
cboFunctions.Items.Add("COLUMNS(, ,)"); //Возвращает количество столбцов в ссылке (reference) или массиве
cboFunctions.Items.Add("INDEX(, , ,)"); //Возвращает значение или ссылку (reference) на значение из таблицы или диапазона ячеек
cboFunctions.Items.Add("CEILING(, , ,)"); //Возвращает число, округляемое в сторону увеличения, от нуля до ближайшего кратного указанному значению
cboFunctions.Items.Add("LEN(, ,)"); //Возвращает количество символов в текстовой строке
cboFunctions.Items.Add("LENB(, ,)"); //Возвращает количество байт, используемых для представления знаков в текстовой строке
cboFunctions.Items.Add("ROUND(, , ,)"); //Округление числа до ближайшего значения. Примеры: ROUND(B8), ROUND(E8, 2)
```

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

### 1. Сбрасываем таблицу - это пригодится при открытии нового документа

```
'VB
rgrid.CurrentWorksheet.Reset()

//C#
rgrid.CurrentWorksheet.Reset();
```

### 2. Загрузка данных. Здесь по 5 вариантов загрузки как из "фирменного" формата, так и из csv

#### 2.1 Просто указываем имя файла

```
'VB
rGrid.Load(Application.StartupPath & "\\test.xml")
//C#
rGrid.Load(Application.StartupPath + "\\test.xml");
```

## 2.2 Указание имени файла и кодировки

```
'VB
rGrid.Load(Application.StartupPath & "\\test.xml", Encoding.UTF8)
//C#
rGrid.Load(Application.StartupPath + "\\test.xml", Encoding.UTF8);
```

## 2.3-2.5 Работа с потоком ввода-вывода

```
'VB
Using ss As FileStream = New FileStream(Application.StartupPath & "\\test.xml", FileMode.Create, FileAccess.Read)
'Код одного из пунктов 2.3-2.5
End Using
//C#
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.xml", FileMode.Create, FileAccess.Read)) {
//Код одного из пунктов 2.3-2.5
}
```

## 2.3 Указание потока

```
'VB
rGrid.Load(ss)
//C#
rGrid.Load(ss);
```

## 2.4 Ещё вариант

```
'VB
Dim rr As System.Func(Of String, System.Type)
rGrid.Load(ss, rr)
//C#
System.Func rr = default(System.Func);
rGrid.Load(ss, rr);
```

## 2.5 Указание потока, формата файла

```
'VB
Dim ff As unvell.ReoGrid.IO.FileFormat = unvell.ReoGrid.IO.FileFormat.ReoGridXML
rGrid.Load(ss, ff)
//C#
unvell.ReoGrid.IO.FileFormat ff = unvell.ReoGrid.IO.FileFormat.ReoGridXML;
rGrid.Load(ss, ff);
```

## Загрузка данных CSV

### 2.6 Просто указываем имя файла

```
'VB
rgrid.CurrentWorksheet.LoadCSV(Application.StartupPath & "\\test.csv")
//C#
rgrid.CurrentWorksheet.LoadCSV(Application.StartupPath + "\\test.csv");
```

### 2.7 Указание имени файла и кодировки

```
'VB
rGrid.CurrentWorksheet.LoadCSV(Application.StartupPath & "\\test.csv", Encoding.UTF8)
//C#
rGrid.CurrentWorksheet.LoadCSV(Application.StartupPath + "\\test.csv", Encoding.UTF8);
```

### 2.8-2.10. Работа с потоком ввода-вывода

```
'VB
Using ss As FileStream = New FileStream(Application.StartupPath & "\\test.csv", FileMode.Create, FileAccess.Read)
'Код одного из пунктов 2.8-2.10
End Using
//C#
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.csv", FileMode.Create, FileAccess.Read)) {
//Код одного из пунктов 2.8-2.10
}
```

### 2.8 Указание потока

```
'VB
rGrid.CurrentWorksheet.LoadCSV(ss)
//C#
rGrid.CurrentWorksheet.LoadCSV(ss);
```

### 2.9 Указание потока и кодировки

```
'VB
rGrid.CurrentWorksheet.LoadCSV(ss, Encoding.UTF8)
//C#
rGrid.CurrentWorksheet.LoadCSV(ss, Encoding.UTF8);
```

### 2.10 Указание потока, кодировки и дополнительных параметров

```
'VB
rGrid.CurrentWorksheet.LoadCSV(ss, True, 10, Encoding.UTF8)
//C#
```

```
rGrid.CurrentWorksheet.LoadCSV(ss, true, 10, Encoding.UTF8);
```

## 2.11 С диалогом открытия файла. На проверках сэкономили

```
'VB
Dim dlgOpen As New OpenFileDialog
With dlgOpen
    .Title = "Открываем файл"
    .FileName = Application.StartupPath & "\docs"
    .DefaultExt = ".rgf"
    .Filter = "Книга Reogrid (rgf)|*.rgf"
    .ShowDialog()

    rgrid.CurrentWorksheet.Load(.FileName)
End With

//C#
OpenFileDialog dlgOpen = new OpenFileDialog();
var _with1 = dlgOpen;
_with1.Title = "Открываем файл";
_with1.FileName = Application.StartupPath + "\\docs";
_with1.DefaultExt = ".rgf";
_with1.Filter = "Книга Reogrid (rgf)|*.rgf";
_with1.ShowDialog();

rgrid.CurrentWorksheet.Load(_with1.FileName);
```

## 3. Сохранение данных в "фирменный" формат. Доступно 4 способа

### 3.1 С проверкой

```
'VB
rgrid.CurrentWorksheet.Save(Application.StartupPath & "\test.xml")
'Определение результата
If rgrid.CurrentWorksheet.Save(Application.StartupPath & "\test.xml") = True Then MsgBox("Сохранение успешно произведено!")

//C#
rgrid.CurrentWorksheet.Save(Application.StartupPath + "\test.xml");
//Определение результата
if (rgrid.CurrentWorksheet.Save(Application.StartupPath + "\test.xml") == true)
    Interaction.MsgBox("Сохранение успешно произведено!");
```

### 3.2 С "подписью"

```
'VB
rgrid.CurrentWorksheet.Save(Application.StartupPath & "\test.xml", "MyEditor")

//C#
rgrid.CurrentWorksheet.Save(Application.StartupPath + "\test.xml", "MyEditor");
```

Для пунктов 3.3 и 3.4 получим имя файлового потока

```
'VB
Using ss As FileStream = New FileStream(Application.StartupPath & "\test.xml", FileMode.Create, FileAccess.Write)
    ' Сюда вставляем код из пунктов 3.3 или 3.4
End Using

//C#
using (FileStream ss = new FileStream(Application.StartupPath + "\test.xml", FileMode.Create, FileAccess.Write)) {
    // Сюда вставляем код из пунктов 3.3 или 3.4
}
```

### 3.3 Без подписи, с указанием потока вывода

```
'VB
rgrid.CurrentWorksheet.Save(ss)

//C#
rgrid.CurrentWorksheet.Save(ss);
```

### 3.4 С подписью и указанием потока вывода

```
'VB
rgrid.CurrentWorksheet.Save(ss, "MyEditor")

//C#
rgrid.CurrentWorksheet.Save(ss, "MyEditor");
```

## Вариант работы с диалоговым окном

```
'VB
Dim dlgExport As New SaveFileDialog
With dlgExport
    .Title = "Сохраняем в формате rgf"
    .DefaultExt = ".rgf"
    .Filter = "Книга Reogrid (rgf)|*.rgf"
    .FileName = Application.StartupPath
    .ShowDialog()

    rgrid.CurrentWorksheet.Save(.FileName)
End With

//C#
SaveFileDialog dlgExport = new SaveFileDialog();
var _with1 = dlgExport;
_with1.Title = "Сохраняем в формате rgf";
_with1.DefaultExt = ".rgf";
_with1.Filter = "Книга Reogrid (rgf)|*.rgf";
_with1.FileName = Application.StartupPath;
_with1.ShowDialog();

rgrid.CurrentWorksheet.Save(_with1.FileName);
```

## 4. Экспорт в HTML двумя способами

```
'VB
Using ss As FileStream = New FileStream(Application.StartupPath & "\test.html", FileMode.Create, FileAccess.Write)

'1. Со значением заголовка по умолчанию
rGrid.ExportAsHTML(ss)

'2. С указанием заголовка
rGrid.ExportAsHTML(ss, "Заголовок страницы")
End Using
```

```
//C#
using (FileStream ss = new FileStream(Application.StartupPath + "\\test.html", FileMode.Create, FileAccess.Write)) {

//1. Со значением заголовка по умолчанию
rgrid.ExportAsHTML(ss);

//2. С указанием заголовка
rgrid.ExportAsHTML(ss, "Заголовок страницы");
}
}
```

### Пример с диалоговым окном сохранения файла

```
'VB
Dim dlgExport As New SaveFileDialog
With dlgExport
.Title = "Сохраняем в формате html"
.DefaultExt = "html"
.Filter = "Веб-страницы (html)|*.html"
.FileName = Application.StartupPath
.ShowDialog()

Using ss As FileStream = New FileStream(.FileName, FileMode.Create, FileAccess.Write)
rgrid.CurrentWorksheet.ExportAsHTML(ss, "Заголовок страницы")
End Using
End With

//C#
SaveFileDialog dlgExport = new SaveFileDialog();
var _with1 = dlgExport;
_with1.Title = "Сохраняем в формате html";
_with1.DefaultExt = "html";
_with1.Filter = "Веб-страницы (html)|*.html";
_with1.FileName = Application.StartupPath;
_with1.ShowDialog();

using (FileStream ss = new FileStream(_with1.FileName, FileMode.Create, FileAccess.Write)) {
rgrid.CurrentWorksheet.ExportAsHTML(ss, "Заголовок страницы");
}
}
```

## 5. Печать (Printing)

### 5.1 Рассмотрим несколько моментов, связанных с печатью.

#### 5.1.1 Авторазбивка на страницы (Auto Paging)

При выводе содержимого таблицы на печать, её содержимое разделяется между страницами, как в MS Excel.

Для примера выполняем авторазбивку на страницы или сбрасываем область печати

```
'VB
rgrid.CurrentWorksheet.AutoSplitPage()

//C#
rgrid.CurrentWorksheet.AutoSplitPage();
```

**Примечание:** сам этот метод достаточно быстр (t<10 мс), но при использовании удалённого принтера может занять гораздо больше времени. Ниже приведён небольшой пример с использованием локального и удалённого принтеров

Текущий принтер	Затраченное время
Microsoft XPS Document Writer	50 мс
FX DocuCentre-IV C4470 (удалённый принтер)	147 мс

#### 5.1.2 Отображаем границы страниц (область печати)

```
'VB
rgrid.CurrentWorksheet.EnableSettings(ReoGridSettings.View_ShowPageBreaks)

//C#
rgrid.CurrentWorksheet.EnableSettings(ReoGridSettings.View_ShowPageBreaks);
```

#### 5.1.3 Автопроверка границ печати

Если область печати не задана, **ReoGrid** попытается определить её самостоятельно - по содержимому ячеек

#### 5.1.4 Изменение диапазона ячеек для печати

Задаём область печати через свойство **PrintableRange**. В качестве параметра указывается диапазон ячеек. Нумерация начинается с 0.

```
'VB
rgrid.CurrentWorksheet.PrintableRange = New ReoGridRange(1, 1, 9, 9)

//C#
rgrid.CurrentWorksheet.PrintableRange == new ReoGridRange(1, 1, 9, 9);
```

Если печатаемый диапазон не помещается на листе выбранного формата, он будет разбит на несколько страниц.

#### 5.1.5 Изменение порядка следования страниц

```
'VB
rgrid.CurrentWorksheet.PrintSettings.PageOrder = PrintPageOrder.DownThenOver

//C#
rgrid.CurrentWorksheet.PrintSettings.PageOrder == PrintPageOrder.DownThenOver;
```

#### 5.1.6 Изменение настроек страницы

Свойство **PageSettings** используется для изменения настроек бумаги

```
'VB
'Альбомная ориентация листа
rgrid.CurrentWorksheet.PageSettings.Landscape = True

//C#
//Альбомная ориентация листа
rgrid.CurrentWorksheet.PageSettings.Landscape == true;
```

#### 5.1.7 Вставка разрыва страницы

```
'VB
rGrid.ColumnPageBreaks.Add(5)

//C#
rGrid.ColumnPageBreaks.Add(5);
```

### 5.1.8 Обновление разбивки страницы

```
'VB
rgrid.CurrentWorksheet.AutoSplitPage()

//C#
rgrid.CurrentWorksheet.AutoSplitPage();
```

### 5.1.9 Запрещаем пользователю менять настройки

```
'VB
rgrid.CurrentWorksheet.DisableSettings(ReoGridSettings.Behavior_AllowUserChangingPageBreaks)

//C#
rgrid.CurrentWorksheet.DisableSettings(ReoGridSettings.Behavior_AllowUserChangingPageBreaks);
```

## 5.2 Вызов диалога настройки печати

```
'VB
Using psd As PageSetupDialog = New PageSetupDialog()
psd.PageSettings = rgrid.CurrentWorksheet.PageSettings.Clone()

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

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

//C#
using (PageSetupDialog psd = new PageSetupDialog()) {
psd.PageSettings = rgrid.CurrentWorksheet.PageSettings.Clone();

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

if (psd.ShowDialog() == System.Windows.Forms.DialogResult.OK) {
rgrid.CurrentWorksheet.PageSettings = psd.PageSettings.Clone();
rgrid.CurrentWorksheet.ClearAllPageBreaks();
}
}
```

## 5.3 Вариант вызова диалога предварительного просмотра

```
'VB
'Создаём документ для печати
Dim doc = rgrid.CurrentWorksheet.CreatePrintDocument()

'Показываем диалоговое окно предварительного просмотра
Using ppd As PrintPreviewDialog = New PrintPreviewDialog()
ppd.Document = doc
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
{
dynamic doc = rgrid.CurrentWorksheet.CreatePrintDocument();

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

## 5.4 Вариант вывода на печать с использованием стандартного диалогового окна печати

```
'VB
Dim doc As ReoGridPrintDocument = Nothing
Try
doc = rgrid.CurrentWorksheet.CreatePrintDocument()
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
doc.Dispose() 'If doc = Nothing Then Else doc.Dispose()

//C#
ReoGridPrintDocument doc = null;
try {
doc = rgrid.CurrentWorksheet.CreatePrintDocument();
} catch (Exception ex) {
MessageBox.Show(this, ex.Message, this.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Information);
return;
}

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)) {
    doc.PrinterSettings = pd.PrinterSettings;
    doc.Print();
}
}
//If doc = Nothing Then Else doc.Dispose()
doc.Dispose();

```

## 5.5 Сброс разрывов страниц

```

'VB
rgrid.CurrentWorksheet.ResetAllPageBreaks()

//C#
rgrid.CurrentWorksheet.ResetAllPageBreaks();

```

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

### 1. Отмена последнего действия, если возможно

```

'VB
If rgrid.CurrentWorksheet.CanUndo = True Then rgrid.CurrentWorksheet.Undo()

//C#
if (rgrid.CurrentWorksheet.CanUndo == true) rgrid.CurrentWorksheet.Undo();

```

### 2. Возврат отменённого действия, если возможен

```

'VB
If rgrid.CurrentWorksheet.CanRedo = True Then rgrid.CurrentWorksheet.Redo()

//C#
if (rgrid.CurrentWorksheet.CanRedo == true) rgrid.CurrentWorksheet.Redo();

```

### 3. Повтор последнего действия

```

'VB
Try
    rgrid.CurrentWorksheet.RepeatLastAction(rgrid.CurrentWorksheet.SelectionRange)
Catch ex As Exception
    MsgBox(ex.Message)
End Try

//C#
try {
    rgrid.CurrentWorksheet.RepeatLastAction(rgrid.CurrentWorksheet.SelectionRange);
} catch (Exception ex) {
    Interaction.MsgBox(ex.Message);
}

```

### 4. Выбрать все (Select All). Выделяет все ячейки

```

'VB
rgrid.CurrentWorksheet.SelectAll()

//C#
rgrid.CurrentWorksheet.SelectAll();

```

## Буфер обмена (Clipboard)

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**

### Вырезать (Cut). Проверяем возможность и выполняем процедуру

```

'VB
If rgrid.CurrentWorksheet.CanCut = True Then rgrid.CurrentWorksheet.Cut()

//C#
if (rgrid.CurrentWorksheet.CanCut == true) rgrid.CurrentWorksheet.Cut();

```

Для события **BeforeCut** можно выполнить:

```

'VB
'Перехват горячих клавиш
e.IsCancelled = True

//C#
//Перехват горячих клавиш
e.IsCancelled == true

```

### Копировать (Copy)

```

'VB
If rgrid.CurrentWorksheet.CanCopy = True Then rgrid.CurrentWorksheet.Copy()

//C#
if (rgrid.CurrentWorksheet.CanCopy == true) rgrid.CurrentWorksheet.Copy();

```

Для события **BeforeCopy**:

```

'VB
'Перехват горячих клавиш
e.IsCancelled = True

//C#
//Перехват горячих клавиш
e.IsCancelled == true

```

## Вставить (Paste)

```
'VB
If rgrid.CurrentWorksheet.CanPaste = True Then rgrid.CurrentWorksheet.Paste()

//C#
if (rgrid.CurrentWorksheet.CanPaste == true) rgrid.CurrentWorksheet.Paste();
```

Если нужно контролировать процесс получения данных из буфера обмена:

```
'VB
Dim text As String = Clipboard.GetText()

'Преобразование в массив типа Объект
Dim data(,) As Object = RGUtility.ParseTabbedString(text)

'Перенос данных в таблицу
rgrid.CurrentWorksheet.SetRangeData(rgrid.CurrentWorksheet.SelectionRange, data)

//C#
string text = Clipboard.GetText();

//Преобразование в массив типа Объект
object[,] data = RGUtility.ParseTabbedString(text);

//Перенос данных в таблицу
rgrid.CurrentWorksheet.SetRangeData(rgrid.CurrentWorksheet.SelectionRange, data);
```

Для события **BeforePaste**:

```
'VB
'Перехват горячих клавиш
e.IsCancelled = True

//C#
//Перехват горячих клавиш
e.IsCancelled == true
```

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

### 1. Стиль выделения (Selection Style). Определяет внешний вид активной ячейки

#### 1.1 Отсутствует

```
'VB
rgrid.CurrentWorksheet.SelectionStyle = ReoGridSelectionStyle.None

//C#
rgrid.CurrentWorksheet.SelectionStyle == ReoGridSelectionStyle.None;
```

#### 1.2 По умолчанию

```
'VB
rgrid.CurrentWorksheet.SelectionStyle = ReoGridSelectionStyle.Default

//C#
rgrid.CurrentWorksheet.SelectionStyle == ReoGridSelectionStyle.Default;
```

#### 1.3 Фокус Windows

```
'VB
rgrid.CurrentWorksheet.SelectionStyle = ReoGridSelectionStyle.FocusRect

//C#
rgrid.CurrentWorksheet.SelectionStyle == ReoGridSelectionStyle.FocusRect;
```

### 2. Режим выделения (Selection Mode). Определяет активный элемент (по умолчанию - диапазон).

#### 2.1 Отсутствует

```
'VB
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.None

//C#
rgrid.CurrentWorksheet.SelectionMode == ReoGridSelectionMode.None;
```

#### 2.2 Одна ячейка

```
'VB
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Cell

//C#
rgrid.CurrentWorksheet.SelectionMode == ReoGridSelectionMode.Cell;
```

#### 2.3 Диапазон ячеек

```
'VB
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Range

//C#
rgrid.CurrentWorksheet.SelectionMode == ReoGridSelectionMode.Range;
```

#### 2.4 Строка

```
'VB
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Row

//C#
rgrid.CurrentWorksheet.SelectionMode == ReoGridSelectionMode.Row;
```

#### 2.5 Столбец

```
'VB
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Column

//C#
rgrid.CurrentWorksheet.SelectionMode == ReoGridSelectionMode.Column;
```

### 3. Направление перехода (Direction). Определяет направление перехода фокуса после окончания правки ячейки

#### 3.1 Вправо

```
'VB
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Right
//C#
rgrid.CurrentWorksheet.SelectionForwardDirection == SelectionForwardDirection.Right;
```

#### 3.2 Вниз

```
'VB
rgrid.CurrentWorksheet.SelectionForwardDirection = SelectionForwardDirection.Down
//C#
rgrid.CurrentWorksheet.SelectionForwardDirection == SelectionForwardDirection.Down;
```

Другие параметры

#### 4. Отображение линии сетки (Grid Lines)

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowGridLine, chkViewVisibleGridLines.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowGridLine, chkViewVisibleGridLines.Checked);
```

#### 5. Разрывы страниц (Page Breaks)

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowPageBreaks, chkViewVisiblePageBreaks.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowPageBreaks, chkViewVisiblePageBreaks.Checked);
```

#### 6. Отображение горизонтальной полосы прокрутки

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowHorScroll, chkViewVisibleHorizScrollbar.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowHorScroll, chkViewVisibleHorizScrollbar.Checked);
```

#### 7. Отображение вертикальной полосы прокрутки

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowVerScroll, chkViewVisibleVertScrollbar.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowVerScroll, chkViewVisibleVertScrollbar.Checked);
```

#### 8. Отображение заголовков строк

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowRowHeader, chkViewVisibleRowHeader.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowRowHeader, chkViewVisibleRowHeader.Checked);
```

#### 9. Отображение заголовков столбцов

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowColumnHeader, chkViewVisibleColumnHeader.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_ShowColumnHeader, chkViewVisibleColumnHeader.Checked);
```

#### 10. Отображение внешних границ строк, если они существуют

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_AllowShowRowOutlines, chkViewVisibleEnableRowOutlinePanel.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_AllowShowRowOutlines, chkViewVisibleEnableRowOutlinePanel.Checked);
```

#### 11. Отображение внешних границ столбцов, если они существуют

```
'VB
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_AllowShowColumnOutlines, chkViewVisibleEnableColumnOutlinePanel.Checked)
//C#
rgrid.CurrentWorksheet.SetSettings(ReoGridSettings.View_AllowShowColumnOutlines, chkViewVisibleEnableColumnOutlinePanel.Checked);
```

#### 12. Изменение размеров таблицы (Resize)

```
'VB
Dim intRows As Integer = rgrid.CurrentWorksheet.RowCount
Dim intCols As Integer = rgrid.CurrentWorksheet.ColumnCount

intRows = Val(TextBox("Укажите число строк", "Новое число строк", intRows.ToString).Trim)
intCols = Val(TextBox("Укажите число столбцов", "Новое число столбцов", intCols.ToString).Trim)
rgrid.CurrentWorksheet.Resize(intRows, intCols)

//C#
int intRows = rgrid.CurrentWorksheet.RowCount;
int intCols = rgrid.CurrentWorksheet.ColumnCount;

intRows = Conversion.Val(Interaction.TextBox("Укажите число строк", "Новое число строк", intRows.ToString).Trim);
intCols = Conversion.Val(Interaction.TextBox("Укажите число столбцов", "Новое число столбцов", intCols.ToString).Trim);
rgrid.CurrentWorksheet.Resize(intRows, intCols);
```

## Работа со шрифтом и содержимым ячеек

### 1. Шрифт (Font). Гарнитура, кегль, цвет.



### 1.1 Меняем гарнитуру (семейство, имя). Пригодится для события TextChanged списка cboFontName

```
'VB
'Изменяем имя шрифта (гарнитуру)
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontName
MyStyleFont.FontName = cboFontName.Text
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
//Изменяем имя шрифта (гарнитуру)
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontName;
MyStyleFont.FontName = cboFontName.Text;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

### 1.2 Меняем кегль (размер). Пригодится для события TextChanged списка cboFontSize

```
'VB
'Изменяем размер (кегель) шрифта
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontSize
MyStyleFont.FontSize = Val(cboFontSize.Text)
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
//Изменяем размер (кегель) шрифта
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontSize;
MyStyleFont.FontSize = Conversion.Val(cboFontSize.Text);
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

#### 1.2.1 Увеличиваем кегль (размер) шрифта на 1 пт

```
'VB
Dim MyStyle As New ReoGridStyleObject
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange)

'Изменяем размер (кегель) шрифта
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontSize
MyStyleFont.FontSize = MyStyle.FontSize + 1
cboFontSize.Text = MyStyleFont.FontSize
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyle = new ReoGridStyleObject();
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange);

//Изменяем размер (кегель) шрифта
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontSize;
MyStyleFont.FontSize = MyStyle.FontSize + 1;
cboFontSize.Text = MyStyleFont.FontSize;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

#### 1.2.2 Уменьшаем кегль (размер) шрифта на 1 пт

```
'VB
Dim MyStyle As New ReoGridStyleObject
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange)

'Изменяем размер (кегель) шрифта
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontSize
MyStyleFont.FontSize = MyStyle.FontSize - 1
cboFontSize.Text = MyStyleFont.FontSize
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyle = new ReoGridStyleObject();
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange);

//Изменяем размер (кегель) шрифта
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontSize;
MyStyleFont.FontSize = MyStyle.FontSize - 1;
cboFontSize.Text = MyStyleFont.FontSize;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

### 1.3 Цвет текста

```
'VB
'Изменяем цвет шрифта на красный
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.TextColor
MyStyleFont.TextColor = Color.Red
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
//Изменяем цвет шрифта на красный
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.TextColor;
MyStyleFont.TextColor = Color.Red;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

### 1.4 Полу жирный (для пунктов 5.4-5.8 требуется кнопка-флажок с двумя положениями)

```
'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontStyleBold
MyStyleFont.Bold = chkFontBold.Checked
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontStyleBold;
MyStyleFont.Bold = chkFontBold.Checked;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);
```

### 1.5 Курсив (наклонный)

```
'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontStyleItalic
```

```

MyStyleFont.Italic = chkFontItalic.Checked
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontStyleItalic;
MyStyleFont.Italic = chkFontItalic.Checked;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);

```

## 1.6 Подчёркнутый

```

'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontStyleUnderline
MyStyleFont.Underline = chkFontUnderline.Checked
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontStyleUnderline;
MyStyleFont.Underline = chkFontUnderline.Checked;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);

```

## 1.7 Зачёркнутый

```

'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.FontStyleStrikethrough
MyStyleFont.Strikethrough = chkFontStrike.Checked
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.FontStyleStrikethrough;
MyStyleFont.Strikethrough = chkFontStrike.Checked;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);

```

## 1.8 Перенос текста

Устанавливаем режим переноса слов 'WordWrap': (по умолчанию - без переноса, - no-wrap)

```

'VB
Dim MyStyleWrap As New ReoGridStyleObject
MyStyleWrap.Flag = PlainStyleFlag.TextWrap
If chkTextWrap.Checked = True Then MyStyleWrap.TextWrapMode = TextWrapMode.WordBreak Else MyStyleWrap.TextWrapMode = TextWrapMode.NoWrap
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleWrap)

//C#
ReoGridStyleObject MyStyleWrap = new ReoGridStyleObject();
MyStyleWrap.Flag = PlainStyleFlag.TextWrap;
if (chkTextWrap.Checked == true)
    MyStyleWrap.TextWrapMode = TextWrapMode.WordBreak;
else
    MyStyleWrap.TextWrapMode = TextWrapMode.NoWrap;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleWrap);

```

## 2. Выравнивание (Align)

Текст может выравниваться как в горизонтальном, так и в вертикальном направлениях

### 2.1 горизонтальное выравнивание

Для горизонтального выравнивания результат зависит от значения свойства **MyStyleFont.HAlign**

**ReoGridHorAlign.Left** - выравнивание по левому краю

**ReoGridHorAlign.Center** - выравнивание по центру

**ReoGridHorAlign.Right** - выравнивание по правому краю

**ReoGridHorAlign.DistributedIndent** - распределённый отступ

Пример задания горизонтального выравнивания по левому краю

```

'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.HorizontalAlign
MyStyleFont.HAlign = ReoGridHorAlign.Left 'Меняется только эта строка
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.HorizontalAlign;
MyStyleFont.HAlign = ReoGridHorAlign.Left; //Меняется только эта строка
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);

```

### 2.2 Вертикальное выравнивание

Для вертикального выравнивания результат зависит от значения свойства **MyStyleFont.VAlign**

**ReoGridVerAlign.Top** - выравнивание по верхней стороне

**ReoGridVerAlign.Middle** - выравнивание по центру

**ReoGridVerAlign.Bottom** - выравнивание по нижней стороне

Пример задания вертикального выравнивания по верхней стороне

```

'VB
Dim MyStyleFont As New ReoGridStyleObject
MyStyleFont.Flag = PlainStyleFlag.VerticalAlign
MyStyleFont.VAlign = ReoGridVerAlign.Top 'Меняется только эта строка
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont)

//C#
ReoGridStyleObject MyStyleFont = new ReoGridStyleObject();
MyStyleFont.Flag = PlainStyleFlag.VerticalAlign;
MyStyleFont.VAlign = ReoGridVerAlign.Top;
//Меняется только эта строка
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyleFont);

```

## Фиксация ячеек (Freezing)

## 1. Получаем позицию заморозки/закрепления:

```
'VB
Dim pos As ReoGridPos = rgrid.CurrentWorksheet.GetFreezePos()

//C#
ReoGridPos pos = rgrid.CurrentWorksheet.GetFreezePos();
```

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

## 2. Проверяем - зафиксирована ли данная таблица:

```
'VB
Dim isFrozen As Boolean = rgrid.CurrentWorksheet.IsFrozen()

//C#
bool isFrozen = rgrid.CurrentWorksheet.IsFrozen();
```

## 3. Фиксируем до указанной позиции

Вводим третий аргумент, указывающий точку фиксации (в примере - правый нижний край)

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(5, 5, FreezePosition.RightBottom)

//C#
rgrid.CurrentWorksheet.FreezeToCell(5, 5, FreezePosition.RightBottom);
```

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

### 3.1 Обычная фиксация (указана только позиция)

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(5, 0)

//C#
rgrid.CurrentWorksheet.FreezeToCell(5, 0);
```

### 3.2 С привязкой по левому краю

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Left)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Left);
```

### 3.3 С привязкой по правому краю

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Right)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Right);
```

### 3.4 С привязкой по верхнему краю

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Top)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Top);
```

### 3.5 С привязкой по нижнему краю

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Bottom)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.Bottom);
```

### 3.6 С привязкой по левому и верхнему краям

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.LeftTop)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.LeftTop);
```

### 3.7 С привязкой по левому и нижнему краям

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.LeftBottom)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.LeftBottom);
```

### 3.8 С привязкой по правому и верхнему краям

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.RightTop)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.RightTop);
```

### 3.9 С привязкой по правому и нижнему краям

```
'VB
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.RightBottom)

//C#
rgrid.CurrentWorksheet.FreezeToCell(1, 1, FreezePosition.RightBottom);
```

## 4. Отмена фиксации

```
'VB
```

```
rgrid.CurrentWorksheet.Unfreeze ()
//C#
rgrid.CurrentWorksheet.Unfreeze ();
```

## Объединение ячеек и группировка

Немного об особенностях объединения ячеек. Так как **Reogrid** обычно работает с выделенными диапазонами ячеек, то нужно предусматривать проверку: выбрана одна ячейка или более одной.

### 1. Объединение ячеек

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

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

### 2. 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 ReoGridRange(2, 2, 5, 5))
    rgrid.CurrentWorksheet.MergeRange(New ReoGridRange(3, 3, 5, 5))
Catch 'unvell.reogrid.RangeIntersectionException
    MsgBox.Show("Обнаружено исключение RangeIntersectionException! Невозможно изменить фрагмент диапазона ячеек.")
End Try

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

### 3. Разбиение (разделение) ячеек

#### 3.1 Вариант 1. Указанный диапазон

```
'VB
rgrid.CurrentWorksheet.UnmergeRange(New ReoGridRange(0, 0, 10, 10))

//C#
rgrid.CurrentWorksheet.UnmergeRange(new ReoGridRange(0, 0, 10, 10));
```

#### 3.2 Вариант 2. Указанный диапазон

```
'VB
rgrid.CurrentWorksheet.DoAction(New RGUnmergeRangeAction(New ReoGridRange(0, 0, 4, 6)))

//C#
rgrid.CurrentWorksheet.DoAction(new RGUnmergeRangeAction(new ReoGridRange(0, 0, 4, 6)));
```

#### 3.3 Вариант 3. Выделенный диапазон

```
'VB
rgrid.CurrentWorksheet.UnmergeRange(rgrid.CurrentWorksheet.SelectionRange)

//C#
rgrid.CurrentWorksheet.UnmergeRange(rgrid.CurrentWorksheet.SelectionRange);
```

**Примечание.** Метод **UnmergeRange** находит и разбивает все объединённые ячейки указанного диапазона.

## 4. Группировка

### 4.1 Группировка строк

```
'VB
Dim intRow As Integer = 1
Dim intCount As Integer = 3
rgrid.CurrentWorksheet.GroupRows(intRow, intCount)

//C#
int intRow = 1;
int intCount = 3;
rgrid.CurrentWorksheet.GroupRows(intRow, intCount);
```

### 4.2 Группировка столбцов

```
'VB
Dim intCol As Integer = 1
Dim intCount As Integer = 3
rgrid.CurrentWorksheet.GroupColumns(intCol, intCount)

//C#
int intCol = 1;
int intCount = 3;
rgrid.CurrentWorksheet.GroupColumns(intCol, intCount);
```

### 4.3 Разгруппировка строк

```
'VB
Dim intRow As Integer = 1
Dim intCount As Integer = 3
rgrid.CurrentWorksheet.UngroupRows(intRow, intCount)

//C#
int intRow = 1;
int intCount = 3;
rgrid.CurrentWorksheet.UngroupRows(intRow, intCount);
```

### 4.4 Разгруппировка столбцов

```
'VB
Dim intCol As Integer = 1
Dim intCount As Integer = 3
rgrid.CurrentWorksheet.UngroupColumns(intCol, intCount)

//C#
int intCol = 1;
int intCount = 3;
rgrid.CurrentWorksheet.UngroupColumns(intCol, intCount);
```

## Типы ячеек (Cell Types)

Ячейка таблицы может быть дополнительно представлена активным элементом управления.

### 1. Кнопкой (Button)

Задать или изменить надпись на кнопке можно следующим образом:

```
'VB
rgrid.CurrentWorksheet(1, 2) = New ButtonCell()
rgrid.CurrentWorksheet(1, 2) = "Привет"

'Кнопка с обработчиком события нажатия (Click) может выглядеть так:
'Dim btn As ButtonCell = New ButtonCell("Привет!")
'rgrid.CurrentWorksheet(1, 2) = New Object() {"Кнопка: ", btn}
'AddHandler btn.Click, AddressOf btn_Click
'Sub btn_Click(ByVal s, ByVal e)
'MsgBox("Кнопка нажата")
'End Sub

//C#
rgrid.CurrentWorksheet(1, 2) = new ButtonCell();
rgrid.CurrentWorksheet(1, 2) = "Привет";

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

//btn.Click += btn_Click;
/*public void btn_Click(s, e)
{
    Interaction.MsgBox("Кнопка нажата");
}*/
```

### 2. Гиперссылкой (Hyperlink)

Гиперссылка использует данные ячейки как адрес URL.

```
'VB
rgrid.CurrentWorksheet(rgrid.CurrentWorksheet.SelectionRange) = New HyperlinkCell("http://www.yandex.ru")
'или
rgrid.CurrentWorksheet(1, 2) = New HyperlinkCell()
rgrid.CurrentWorksheet(1, 2) = "http://www.yandex.ru"

//C#
rgrid.CurrentWorksheet(rgrid.CurrentWorksheet.SelectionRange) = new HyperlinkCell("http://www.yandex.ru");
//или
rgrid.CurrentWorksheet(1, 2) = new HyperlinkCell();
rgrid.CurrentWorksheet(1, 2) = "http://www.yandex.ru";
```

Если по щелчку на гиперссылке должен происходить переход, то можно применить приведенный ниже код. Объявление переменной-гиперссылки может потребоваться перенести в область объявления переменной на уровне класса формы

```
'VB
Dim link = New HyperlinkCell("http://www.yandex.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#
dynamic link = new HyperlinkCell("http://www.yandex.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));
}*/
```

### 3. Флажком (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 ReoGridStyleObject
middleStyle.Flag = PlainStyleFlag.Padding
middleStyle.Flag = PlainStyleFlag.HorizontalAlign
middleStyle.Padding = New Padding(2)
middleStyle.HAlign = ReoGridHorAlign.Center
Dim grayTextStyle = New ReoGridStyleObject
grayTextStyle.Flag = PlainStyleFlag.TextColor
grayTextStyle.TextColor = Color.DimGray

Dim checkbox As CheckBoxCell = New CheckBoxCell()
rgrid.CurrentWorksheet.SetRangeStyle(7, 2, 1, 1, middleStyle)
rgrid.CurrentWorksheet.SetRangeStyle(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.CheckedChanged, AddressOf checkbox_CheckedChanged
```

```
'Sub checkbox_CheckChanged(ByVal sender, ByVal e)
'MsgBox("Флажок переведён в состояние " & checkbox.ButtonState.ToString())
'End Sub

//C#
dynamic middleStyle = new ReoGridStyleObject();
middleStyle.Flag = PlainStyleFlag.Padding;
middleStyle.Flag = PlainStyleFlag.HorizontalAlign;
middleStyle.Padding = new Padding(2);
middleStyle.HAlign = ReoGridHorAlign.Center;
dynamic grayTextStyle = new ReoGridStyleObject();
grayTextStyle.Flag = PlainStyleFlag.TextColor;
grayTextStyle.TextColor = Color.DimGray;

CheckBoxCell checkbox = new CheckBoxCell();
rgrid.CurrentWorksheet.SetRangeStyle(7, 2, 1, 1, middleStyle);
rgrid.CurrentWorksheet.SetRangeStyle(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());
}*/
```

#### 4. Радиокнопкой (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";
```

#### 5. Раскрывающимся списком (Drop-down List)

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

```
'VB
Dim dropdown As DropdownCell = New DropdownCell(New Object() {"Яблоко", "Апельсин", "Банан", "Груша", "Тыква", "Вишня", "Кокос"})
rgrid.CurrentWorksheet(3, 1) = New Object() {"Список:", dropdown}
rgrid.CurrentWorksheet.SetRangeBorder(3, 2, 1, 3, ReoGridBorderPos.Outline, ReoGridBorderStyle.SolidGray)

//C#
DropdownCell dropdown = new DropdownCell(new object[] {"Яблоко", "Апельсин", "Банан", "Груша", "Тыква", "Вишня", "Кокос"});
rgrid.CurrentWorksheet(3, 1) = new object[] {"Список:", dropdown};
rgrid.CurrentWorksheet.SetRangeBorder(3, 2, 1, 3, ReoGridBorderPos.Outline, ReoGridBorderStyle.SolidGray);
```

#### 6. Картинкой (Image)

Варианты получения картинки для этого типа ячейки

##### 1. Загрузка из файла

```
'VB
Dim image As Bitmap = Bitmap.FromFile("D:\myinfo\photo.png")
rgrid.CurrentWorksheet(1, 2) = New ImageCell(image)

//C#
Bitmap image = Bitmap.FromFile("D:\myinfo\photo.png");
rgrid.CurrentWorksheet(1, 2) = new ImageCell(image);
```

##### 2. Получение из ресурсов

```
'VB
rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.photo)
'или
rgrid.CurrentWorksheet("B3") = New ImageCell(My.Resources.photo)

//C#
rgrid.CurrentWorksheet(1, 2) = new ImageCell(My.Resources.photo);
//или
rgrid.CurrentWorksheet("B3") = new ImageCell(My.Resources.photo);
```

### 3. Ваш вариант ;-)

Свойство `ImageCell ViewMode` позволяет управлять отображением картинки в ячейке:

Создаём `ImageCell` с указанием параметра отображения

```
'VB
Dim img As Bitmap = Bitmap.FromFile("D:\myinfo\photo.png")
Dim imgCell As ImageCell = New ImageCell(img, ImageCellViewMode.Zoom)
rgrid.CurrentWorksheet("B3") = imgCell

//C#
Bitmap img = Bitmap.FromFile("D:\myinfo\photo.png");
ImageCell imgCell = new ImageCell(img, ImageCellViewMode.Zoom);
rgrid.CurrentWorksheet("B3") = imgCell;
```

Также можно указать значение напрямую

```
'VB
imgCell.ViewMode = ImageCellViewMode.Clip

//C#
imgCell.ViewMode == ImageCellViewMode.Clip;
```

И даже можно задать параметры выравнивания по вертикали и горизонтали:

```
'VB
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.PasteHS, ImageCellViewMode.Clip)
'rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.PasteHS, ImageCellViewMode.Stretch)
'rgrid.CurrentWorksheet(1, 2) = New ImageCell(My.Resources.PasteHS, ImageCellViewMode.Zoom)

//C#
dynamic 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.PasteHS, ImageCellViewMode.Clip);
//rgrid.CurrentWorksheet(1, 2) = new ImageCell(My.Resources.PasteHS, ImageCellViewMode.Stretch);
//rgrid.CurrentWorksheet(1, 2) = new ImageCell(My.Resources.PasteHS, ImageCellViewMode.Zoom);
```

## Границы (Borders) и фон ячеек

### 1. Параметры границ диапазона или ячейки

Тип границы задаётся перечислением `ReoGridBorderPos` и может принимать значения:

1.1 Все границы / All Borders 1.2 Внешние границы / Outline 1.3 Внутренние границы / Inside 1.4 Верхняя и нижняя границы (Top-Bottom) 1.5 Левая и правая границы (Left-Right) 1.6 Верхняя граница (Top) 1.7 Нижняя граница (Bottom) 1.8 Левая граница (Left) 1.9 Правая граница (Right) 1.10 Slash (/). Заметить этот вариант не удалось :-). 1.11 Backslash (\). Заметить этот вариант не удалось :-)

### 2. Тип линий

Тип линии задаётся перечислением `BorderLineStyle` и может принимать значения:

2.1 Сплошные (Solid) 2.2 Сплошные полужирные (BoldSolid) 2.3 ...

### 3. Цвет линий

Цвет линии задаётся перечислением `Color` или другим предусмотренным в .Net способом. Ниже приведён пример задания типа линии, цвета для различных вариантов задания границ

```
'VB
Dim MyStyleBorder1 As New ReoGridBorderStyle
MyStyleBorder1.Style = BorderLineStyle.Solid
MyStyleBorder1.Color = Color.Black
rgrid.CurrentWorksheet.SetRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All, MyStyleBorder1)

//C#
ReoGridBorderStyle MyStyleBorder1 = new ReoGridBorderStyle();
MyStyleBorder1.Style = BorderLineStyle.Solid;
MyStyleBorder1.Color = Color.Black;
rgrid.CurrentWorksheet.SetRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All, MyStyleBorder1);
```

### 4. Очистка границ (Clear)

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

```
'VB
Dim MyStyleBorder1 As New ReoGridBorderStyle
MyStyleBorder1.Color = Color.Empty
rgrid.CurrentWorksheet.SetRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All, MyStyleBorder1)

'или

rgrid.CurrentWorksheet.RemoveRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All)

//C#
ReoGridBorderStyle MyStyleBorder1 = new ReoGridBorderStyle();
MyStyleBorder1.Color = Color.Empty;
rgrid.CurrentWorksheet.SetRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All, MyStyleBorder1);

//или

rgrid.CurrentWorksheet.RemoveRangeBorder(rgrid.CurrentWorksheet.SelectionRange, ReoGridBorderPos.All);
```

## 5. Заливка фона

### 5.1 Заливка цветом

```
'VB
Dim MyStyle As New ReoGridStyleObject
MyStyle.Flag = PlainStyleFlag.BackColor
MyStyle.BackColor = Color.Empty 'Очищаем ячейки
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyle)

//C#
```

```

ReoGridStyleObject MyStyle = new ReoGridStyleObject();
MyStyle.Flag = PlainStyleFlag.BackColor;
MyStyle.BackColor = Color.Empty; //Очищаем ячейки
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStyle);

```

## 5.2 Заливка по шаблону

```

'VB
Dim MyStylePattern As New ReoGridStyleObject
MyStylePattern.Flag = PlainStyleFlag.FillPattern
MyStylePattern.Flag = PlainStyleFlag.BackColor
MyStylePattern.BackColor = Color.LightYellow
MyStylePattern.FillPatternColor = Color.SkyBlue
MyStylePattern.FillPatternStyle = System.Drawing.Drawing2D.HatchStyle.DiagonalBrick
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStylePattern)

//C#
ReoGridStyleObject MyStylePattern = new ReoGridStyleObject();
MyStylePattern.Flag = PlainStyleFlag.FillPattern;
MyStylePattern.Flag = PlainStyleFlag.BackColor;
MyStylePattern.BackColor = Color.LightYellow;
MyStylePattern.FillPatternColor = Color.SkyBlue;
MyStylePattern.FillPatternStyle = System.Drawing.Drawing2D.HatchStyle.DiagonalBrick;
rgrid.CurrentWorksheet.SetRangeStyle(rgrid.CurrentWorksheet.SelectionRange, MyStylePattern);

```

## Масштабирование (Zoom in/out)

### 1. Выбор значения масштаба из списка (событие TextChanged для раскрывающегося списка cboZoom)

```

'VB
rgrid.CurrentWorksheet.ScaleFactor = Convert.ToSingle(Val(cboZoom.Text.Replace("%", String.Empty).Trim)) / 100.0F

//C#
rgrid.CurrentWorksheet.ScaleFactor == Convert.ToSingle(Conversion.Val(cboZoom.Text.Replace("%", string.Empty).Trim)) / 100.0F;

```

### 2. Увеличение масштаба на 10%

```

'VB
rgrid.CurrentWorksheet.ZoomIn()

//C#
rgrid.CurrentWorksheet.ZoomIn();

```

### 3. уменьшение масштаба на 10%

```

'VB
rgrid.CurrentWorksheet.ZoomOut()

//C#
rgrid.CurrentWorksheet.ZoomOut();

```

### 4. Сброс масштаба к значению 100%

```

'VB
rgrid.CurrentWorksheet.ZoomReset()

//C#
rgrid.CurrentWorksheet.ZoomReset();

```

### 5. Установка единицы масштаба

```

'VB
rgrid.CurrentWorksheet.SetScale(2.0F)

//C#
rgrid.CurrentWorksheet.SetScale(2.0F);

```

Масштабирование может производиться как с помощью вращения колёсика мыши при зажатой клавише CTRL, так и программно

## Получение данных о выбранном диапазоне

Этот код необходим для настройки пользовательского интерфейса при выборе пользователем другой ячейки или диапазона. Данный код, для версии **0.87**, 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 ReoGridStyleObject 'Создаём экземпляр стиля
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange)

cboFontName.Text = MyStyle.FontName 'Гарнитура шрифта
cboFontSize.Text = MyStyle.FontSize 'Керль шрифта (размер)
chkFontBold.Checked = MyStyle.Bold 'Жирный
chkFontItalic.Checked = MyStyle.Italic 'Курсив
chkFontUnderline.Checked = MyStyle.Underline 'Подчёркнутый
chkFontStrike.Checked = MyStyle.Strikethrough 'Зачёркнутый
grpBackgroundColor.BackColor = MyStyle.BackColor 'Цвет фона
'MyStyle.CopyFrom
'MyStyle.FillPatternColor
'MyStyle.FillPatternStyle

'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

```



```

If MyStyle.HAlign = ReoGridHorAlign.DistributedIndent Then
radTextDisributedIndent.Checked = True
Else
radTextDisributedIndent.Checked = False
End If

'MyStyle.HAlign.General
'MyStyle.Padding
grpTextColor.BackColor = MyStyle.TextColor 'Цвет текста
chkTextWrap.Checked = MyStyle.TextWrapMode 'Перенос по словам

'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

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

ReoGridStyleObject MyStyle = new ReoGridStyleObject();
//Создаём экземпляр стиля
MyStyle = rgrid.CurrentWorksheet.GetRangeStyle(rgrid.CurrentWorksheet.SelectionRange);

cboFontName.Text = MyStyle.FontName;
//Гарнитура шрифта
cboFontSize.Text = MyStyle.FontSize;
//Кегль шрифта (размер)
chkFontBold.Checked = MyStyle.Bold;
//Жирный
chkFontItalic.Checked = MyStyle.Italic;
//Курсив
chkFontUnderline.Checked = MyStyle.Underline;
//Подчёркнутый
chkFontStrike.Checked = MyStyle.Strikethrough;
//Зачёркнутый
grpBackgroundColor.BackColor = MyStyle.BackColor;
//Цвет фона
//MyStyle.CopyFrom
//MyStyle.FillPatternColor
//MyStyle.FillPatternStyle

//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;
}

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

//MyStyle.HAlign.General
//MyStyle.Padding
grpTextColor.BackColor = MyStyle.TextColor;
//Цвет текста
chkTextWrap.Checked = MyStyle.TextWrapMode;
//Перенос по словам

//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;
}

```

## Работа с формулами

1. 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;

```

## 2. Переносим формулу в выбранную ячейку. Используется обработчик события **KeyDown** для текстового поля формулы **txtFormula**

```
'VB
if e.KeyCode = Keys.Enter Then
    rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).Formula = txtFormula.Text
    txtFormula.SelectAll()
End If

//C#
if (e.KeyCode == Keys.Enter) {
    rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).Formula = txtFormula.Text;
    txtFormula.SelectAll();
}
```

## 3. Получение информации 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).Display
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 (Strings.Len(strFormula) >= 0) {
        txtFormula.Text = strFormula;
        cboFunctions.Text = rgrid.CurrentWorksheet.Cells(rgrid.CurrentWorksheet.FocusPos).Display;
    }
}
```

## Тройка примеров функций из англоязычной документации

Данный код можно поместить в обработчики событий **Click** кнопок.

### Пример 1. Пределы. Убеждаемся в том, что **Reogrid** справляется с этой нагрузкой

```
'VB
'Выжимаем максимум! / MaximumGridForm
rgrid.CurrentWorksheet.Resize(1048576, 32768) 'Меняем размер и получаем 1048576 строк и 32768 столбцов
rgrid.CurrentWorksheet.MergeRange(1, 1, 1, 6)
rgrid.CurrentWorksheet(1, 1) = "Предельное количество ячеек (1048576 x 32768)"
rgrid.CurrentWorksheet.MergeRange(3, 1, 2, 6)
rgrid.CurrentWorksheet(3, 1) = "Вы можете побаловаться прокруткой, поиграть с масштабом, отредактировать любую ячейку или изменить её размер."

//C#
//Выжимаем максимум! / MaximumGridForm
{
    rgrid.CurrentWorksheet.Resize(1048576, 32768);

    //Меняем размер и получаем 1048576 строк и 32768 столбцов
    rgrid.CurrentWorksheet.MergeRange(1, 1, 1, 6);
    rgrid.CurrentWorksheet(1, 1) = "Предельное количество ячеек (1048576 x 32768)";
    rgrid.CurrentWorksheet.MergeRange(3, 1, 2, 6);
    rgrid.CurrentWorksheet(3, 1) = "Вы можете побаловаться прокруткой, поиграть с масштабом, отредактировать любую ячейку или изменить её размер.";
}
```

### Пример 2. Настраиваемые заголовки

```
'VB
rgrid.CurrentWorksheet.Resize(10, 7) 'Меняем размер и получаем 10 строк и 7 столбцов

'Приводим стиль первого заголовка к типу "Флажок" (Checkbox)
Dim checkBoxHeader = rgrid.CurrentWorksheet.ColumnHeaders(0)
checkBoxHeader.Text = String.Empty
checkBoxHeader.DefaultCellBody = TypeOf(CheckBoxCell)
checkBoxHeader.WidthInPixel = 30 'Ширина заголовка: 30 пикселей
checkBoxHeader.Style.HorizontalAlign = ReoGridHorAlign.Center
checkBoxHeader.Style.Padding = New Padding(3)

'Установка других заголовков
rgrid.CurrentWorksheet.ColumnHeaders(1).Text = "Продукт"
rgrid.CurrentWorksheet.ColumnHeaders(2).Text = "Цена за единицу"
rgrid.CurrentWorksheet.ColumnHeaders(3).Text = "Скидка"
rgrid.CurrentWorksheet.ColumnHeaders(4).Text = "Количество"
rgrid.CurrentWorksheet.ColumnHeaders(5).Text = "Расширенная цена"
rgrid.CurrentWorksheet.ColumnHeaders(6).Text = "Примечание"

'устанавливаем ширину столбцов (1-4)
rgrid.CurrentWorksheet.SetColumnsWidth(1, 5, 100)

'Включение режима выделения только одной ячейки
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Cell

'Определение региона как диапазона записей
Dim recordRange = rgrid.CurrentWorksheet.DefineNamedRange("records", 0, 0, 4, 6)

'Применение формулы с использованием относительного адреса в порядке для подсчёта каждой строки
'Итоговая цена = (цена за единицу - скидка) * количество
Dim extendedPriceFormula = "( INDIRECT(ADDRESS(ROW(), COLUMN()-3)) " & " - INDIRECT(ADDRESS(ROW(), COLUMN()-2)) )" & " * INDIRECT(ADDRESS(ROW(), COLUMN()-1)) )"

'Вводим данные
recordRange.Data = New Object(,) {
    {False, "Процессор", "230", "0", "1", extendedPriceFormula},
    {True, "Системная плата", "120", "5", "1", extendedPriceFormula},
    {False, "Память", "90", "0", "2", extendedPriceFormula},
    {False, "VGA", "310", "0", "1", extendedPriceFormula}
}

'Сбрасываем стиль курсора для ячеек к обычному виду
rgrid.CellsSelectionCursor = Cursors.Default

'Получаем формулу из выбранной ячейки и отображаем её
'AddHandler rgrid.CurrentWorksheet.FocusPosChanged, AddressOf rGrid_FocusPosChanged
'Sub rGrid_FocusPosChanged(sender,e)
'lblStatus.Text = rgrid.CurrentWorksheet.GetCellFormula(rgrid.CurrentWorksheet.FocusPos)
'End Sub

//C#
rgrid.CurrentWorksheet.Resize(10, 7);
//Меняем размер и получаем 10 строк и 7 столбцов
```

```

//Приводим стиль первого заголовка к типу "Флажок" (Checkbox)
dynamic checkBoxHeader = rgrid.CurrentWorksheet.ColumnHeaders(0);
checkBoxHeader.Text = string.Empty;
//checkBoxHeader.DefaultCellBody = TypeOf(CheckBoxCell)
checkBoxHeader.WidthInPixel = 30;
//Ширина заголовка: 30 пикселей
checkBoxHeader.Style.HorizontalAlignment = ReoGridHorAlign.Center;
checkBoxHeader.Style.Padding = new Padding(3);

//Установка других заголовков
rgrid.CurrentWorksheet.ColumnHeaders(1).Text = "Продукт";
rgrid.CurrentWorksheet.ColumnHeaders(2).Text = "Цена за единицу";
rgrid.CurrentWorksheet.ColumnHeaders(3).Text = "Скидка";
rgrid.CurrentWorksheet.ColumnHeaders(4).Text = "Количество";
rgrid.CurrentWorksheet.ColumnHeaders(5).Text = "Расширенная цена";
rgrid.CurrentWorksheet.ColumnHeaders(6).Text = "Примечание";

//устанавливаем ширину столбцов (1-4)
rgrid.CurrentWorksheet.SetColumnsWidth(1, 5, 100);

//Включение режима выделения только одной ячейки
rgrid.CurrentWorksheet.SelectionMode = ReoGridSelectionMode.Cell;

//Определение региона как диапазона записей
dynamic recordRange = rgrid.CurrentWorksheet.DefineNamedRange("records", 0, 0, 4, 6);

//Применение формулы с использованием относительного адреса в порядке для подсчёта каждой строки
//Итоговая цена = (цена за единицу - скидка) * количество
dynamic extendedPriceFormula = "( INDIRECT(ADDRESS(ROW(), COLUMN()-3)) ) + " - INDIRECT(ADDRESS(ROW(), COLUMN()-2)) ) " + " * INDIRECT(ADDRESS(ROW(), COLUMN()-1)) ";

//Вводим данные
recordRange.Data = new object[ + 1, + 1 ] {
    {false, "Процессор", "230", "0", "1", extendedPriceFormula},
    {true, "Системная плата", "120", "5", "1", extendedPriceFormula},
    {false, "Память", "90", "0", "2", extendedPriceFormula},
    {false, "VGA", "310", "0", "1", extendedPriceFormula}
};

//Сбрасываем стиль курсора для ячеек к обычному виду
rgrid.CellsSelectionCursor = Cursors.Default;

//Получаем формулу из выбранной ячейки и отображаем её
/*rgrid.CurrentWorksheet.FocusPosChanged += rGrid_FocusPosChanged;

public void rGrid_FocusPosChanged(sender, e)
{
    lblStatus.Text = rgrid.CurrentWorksheet.GetCellFormula(rgrid.CurrentWorksheet.FocusPos);
}*/

```

### Пример 3. Определение разрешённого для редактирования диапазона

Многие события, происходящие перед основными, поддерживают свойство **IsCancelled**. Если его установить в **True**, ReoGrid отменит соответствующую операцию. Обычно используется для предотвращения правки ячеек или операций сворачивания/разворачивания

```

'VB
'Пример: текст можно править лишь в чётко определённом диапазоне:
Dim editableRange = New ReoGridRange(3, 1, 2, 3)
rgrid.CurrentWorksheet.SetRangeBorder(editableRange, ReoGridBorderPos.Outline, ReoGridBorderStyle.SolidBlack)
rgrid.CurrentWorksheet(2, 1) = "Правка разрешена только для этого диапазона:"

'AddHandler rgrid.CurrentWorksheet.BeforeCellEdit, AddressOf rgrid_BeforeCellEdit
'Sub rgrid_BeforeCellEdit(ByVal sender, ByVal e)
'e.IsCancelled = Not rgrid.CurrentWorksheet(1, 1, 2, 2).Contains(e.Cell.GetPos())
'End Sub

//C#
//Пример: текст можно править лишь в чётко определённом диапазоне:
{
    dynamic editableRange = new ReoGridRange(3, 1, 2, 3);
    rgrid.CurrentWorksheet.SetRangeBorder(editableRange, ReoGridBorderPos.Outline, ReoGridBorderStyle.SolidBlack);
    rgrid.CurrentWorksheet(2, 1) = "Правка разрешена только для этого диапазона:";

    rgrid.CurrentWorksheet.BeforeCellEdit += rgrid_BeforeCellEdit;
}

/*public void rgrid_BeforeCellEdit(sender, e)
{
    e.IsCancelled = !rgrid.CurrentWorksheet(1, 1, 2, 2).Contains(e.Cell.GetPos());
}*/

```