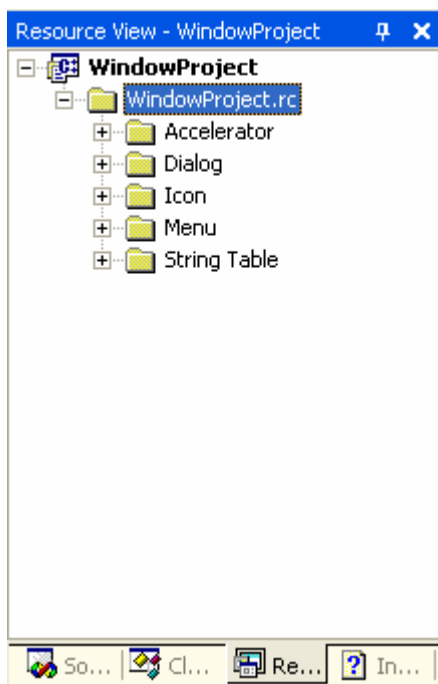


Работа с ресурсами	1
Акселераторы	2
Иконки	3
Окна диалога	5
Static Text	7
Picture Control	9
Button	9
Диалоговое окно	9
Check Box	10
Radio Button	10
Edit Control	12
Combo Box	13
List Box	13
Group Box	14
Horizontal Scroll Box и Vertical Scroll Box	14
Slider control	14
Spin Control	15
Progress Control	15
Hot Key	16
List Control	16
Tree view	17
Tab Control	17
Animation Control	18
Rich Edit	18
Date Time Picker	18
Month Calendar Control	19
IP Address Control	19
Меню	20
Таблицы строк	21
Заключение	22

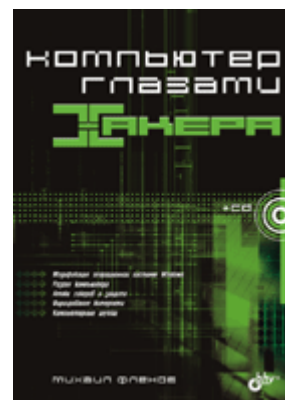
Работа с ресурсами

Для работы с ресурсами в Visual Studio используется окно **Resource View**. Чтобы его отобразить, необходимо выбрать меню **View/Resource View** или можно дважды щелкнуть по файлу ресурсов (файл с расширением .rc) в окне Solution Explorer. Мы будем тестировать ресурсы на основе файла, который вы можете увидеть в папке /doc/resource/WindowProject.rc. Запустите этот файл из проводника, и перед вами откроется окно следующего вида:



Окно состоит из дерева, во главе которого находится элемент с таким же именем, как у проекта. Сразу под ним можно увидеть имя файла ресурса (в данном случае это WindowProject.rc). У нас только один файл, поэтому на этом уровне дерева больше ничего нет. В проекте может быть несколько таких файлов, и все они будут на втором уровне дерева. Следующий уровень – это папки, которые группируют ресурсы по типу. У нас в проекте используются ресурсы следующего типа:

- ☐ Accelerator – здесь находится список быстрого вызова;
- ☐ Dialog – визуальные окна диалога. Да, это окна, которые вы можете создавать визуально;
- ☐ Icon – иконки;
- ☐ Menu – меню;



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

- ❑ String Table – таблицы строк.

Помимо этого могут быть ресурсы следующего типа:

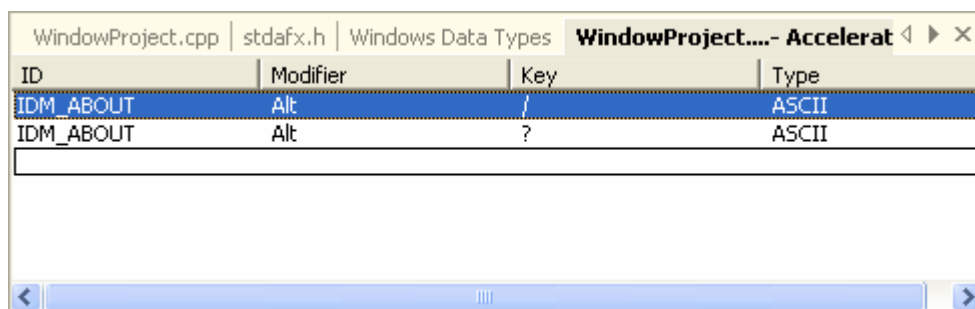
- ❑ Bitmap – битовые рисунки;
- ❑ Cursor – курсоры;
- ❑ Version – информация о версии;
- ❑ HTML – HTML файл.

Давайте посмотрим, как можно редактировать уже существующие ресурсы каждого типа.

Акселераторы

В папке Accelerator окна Resource View у нас есть ресурс с именем IDC_WINDOWPROJECT. Выделите его, и в окне **Properties** появятся его свойства. Если это окно не видно, то выберите меню **View/Properties**. Среди свойств есть name, с помощью которого можно изменить имя ресурса и Language, в котором можно указать язык ресурса.

Чтобы открыть ресурс для редактирования, дважды щелкните по его имени в дереве **Resource View**. Перед вами откроется окно, как на следующем рисунке:



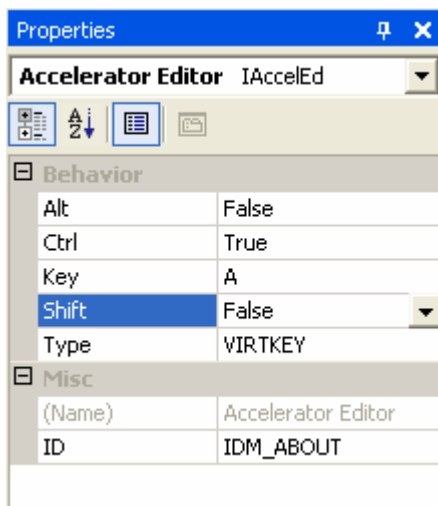
Здесь у нас таблица, которая состоит из четырех колонок:

- ❑ ID – идентификатор ресурса. В этой колонке указываются идентификаторы пунктов меню или других ресурсов (например, кнопок на окнах диалога), которые должны вызываться указанной горячей клавишей. В данном случае это пункт меню **Help/About**, который имеет идентификатор **IDM_ABOUT**;
- ❑ Modifier – модификатор. Здесь указываются клавиши Ctrl, Alt или Shift, если они участвуют в сочетании горячих клавиш;
- ❑ Key – клавиша, которая должна быть нажата совместно с модификатором, если модификатор есть, или отдельно, если модификатора нет;
- ❑ Type – тип клавиши.

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Каждая запись в таблице – это отдельная горячая клавиша. Одному пункту меню может быть назначено несколько горячих клавиш, как в данном случае, где за вывод окна с информацией о программе отвечает клавиши **Ctrl+/** и **Ctrl+?**. Правда если посмотреть на клавиатуру, то вы увидите, что символы / и ? находятся на одной клавише, по крайней мере у меня ☺.

Чтобы изменить клавишу, выделите необходимую строку и посмотрите на окно свойств:



Допустим, что вы хотите, чтобы горячая клавиша была **Ctrl+A**. Для этого, сначала нужно переключить тип клавиши (свойство **Type**) на **VIRTKEY**. Дело в том, что клавиша **Ctrl** может быть в сочетании только при этом типе. Теперь в свойствах **Alt** и **Shift** установите значение **false**, потому что эти клавиши не будут участвовать. А теперь, в свойстве **Ctrl** установите значение **true**. Модификатор **Ctrl** мы установили, теперь нужно указать клавишу **A**. Введите эту букву в свойстве **Key**.

Если вы хотите изменить элемент управления, который должен реагировать на горячую клавишу, то укажите его идентификатор в свойстве **ID**. Чтобы не ошибиться в названии идентификатора, можно выбрать его из выпадающего списка, который появляется, если выделить это свойство.

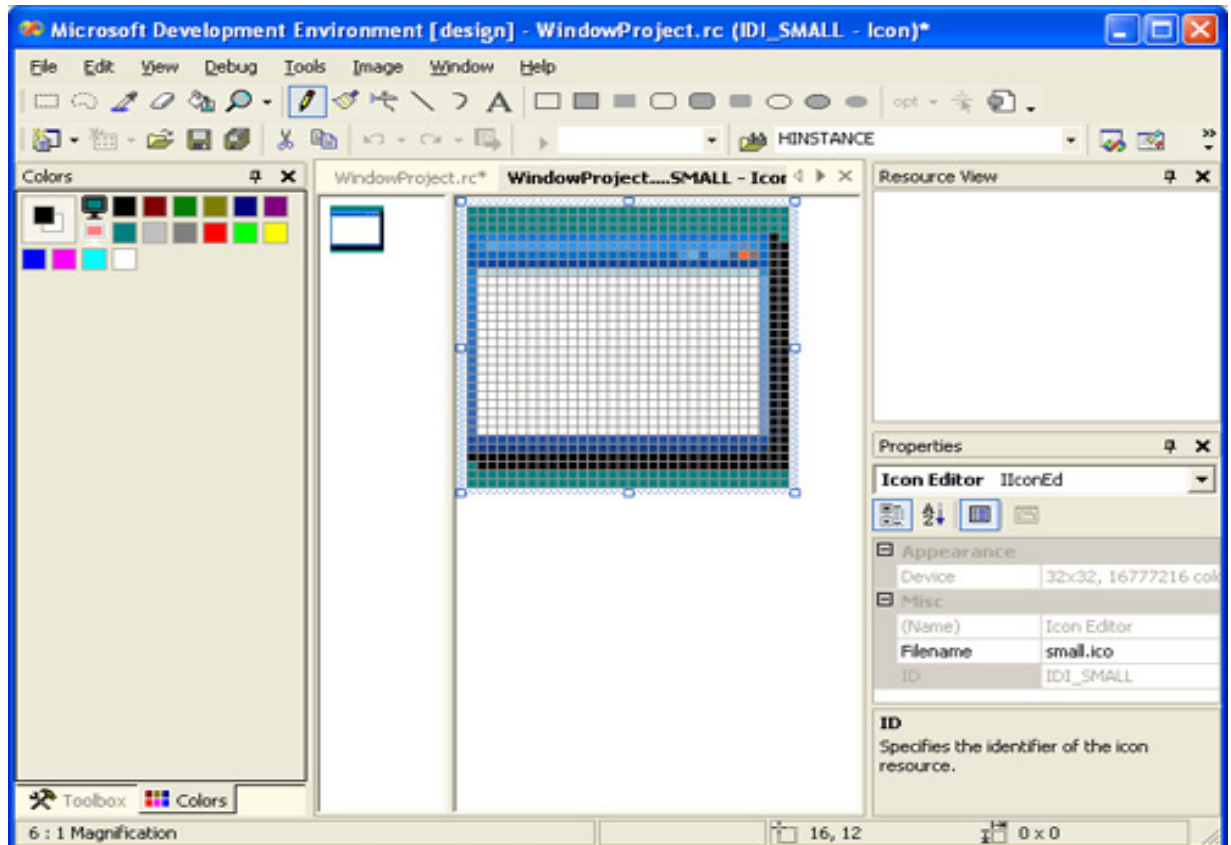
Иконки

В разделе Icon можно увидеть иконки, которые будут отображаться у программы в файловом менеджере и в заголовке программы. В данном случае, у нас две иконки разного размера. Выбирая в дереве ресурсов названия иконок, в окне **Properties** можно увидеть свойство **Filename** – где указывается имя файла, в котором храниться сама иконка. Дважды щелкнув по названию иконки, внешний вид среды разработки Visual Studio меняется на простенький графический редактор, заточенный



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

специально для работы с иконками, как показано на следующем рисунке:



Иконки ограничены в цветах, поэтому они используют палитры. В данном случае можно использовать всего 16 цветов, и все они представлены на панели слева. Центральная часть разделена на две части: слева находится картинка в её реальном размере, а справа – увеличенный вариант, на котором вы можете рисовать.

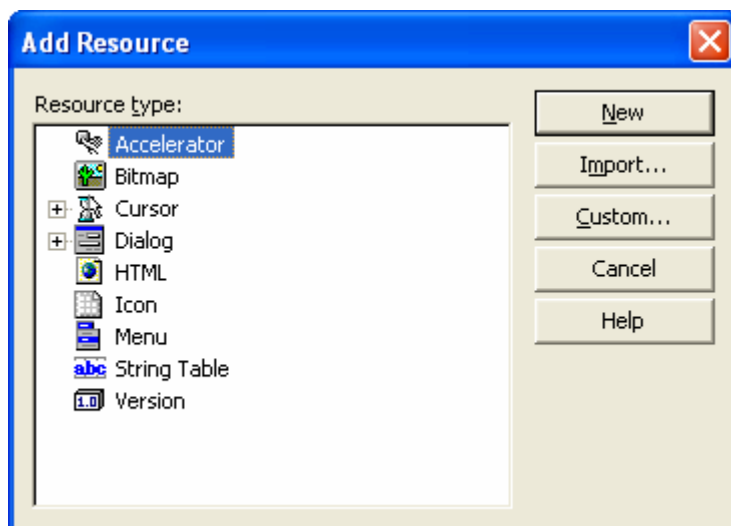
А где же находятся инструменты, с помощью которых можно рисовать? А они находятся на панели сверху окна, сразу под меню. Тут есть карандаш, кисть, прямоугольник, овал и т.д.

Учиться рисовать мы не будем, потому что я сам не художник и рисую, как курица лапой, но посмотрим, что тут ещё есть. А есть у нас меню **Image**. Тут множество различных дополнительных возможностей для переворота изображения, переворота, но самое важное храниться в **Image/Current Icon Image Types**. Здесь находятся различные типы иконок, которые могут отличаться количеством поддерживаемых цветов и размером. Зачем это нужно? Дело в том, что в Windows 95/98 используются иконки в 256 цветов, а в Windows 2000 и XP уже можно использовать полноцветные

Данный документ распространяется только на диск к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

иконки, которые намного красивее. Если в вашей программе будет только иконка в 16 цветов, то она будет корректно отображена в любой системе, но выглядеть не очень красиво, а если полноцветная иконка, то она будет красиво выглядеть в Windows XP, но возникнут проблемы с её отображением в старых версиях Windows.

Давайте попробуем добавить новую иконку и что-нибудь нарисовать. Вернитесь в окно **Resource Explorer** и щелкните в нем правой кнопкой. В появившемся меню выберите пункт **Add Resource** и перед вами появится окно, как на следующем рисунке:



Здесь выбираем пункт **Icon** и нажимаем кнопку **New**. В результате на экране появится пустое окно для создания новой иконки. Нарисуйте что-нибудь. Теперь давайте зададим имя для идентификатора иконки. Для этого сначала закройте окно редактирования иконки, выделите имя созданной иконки в дереве ресурсов и в окне свойств (**Properties**) в свойстве **ID** введите `IDI_MY_ICON`.

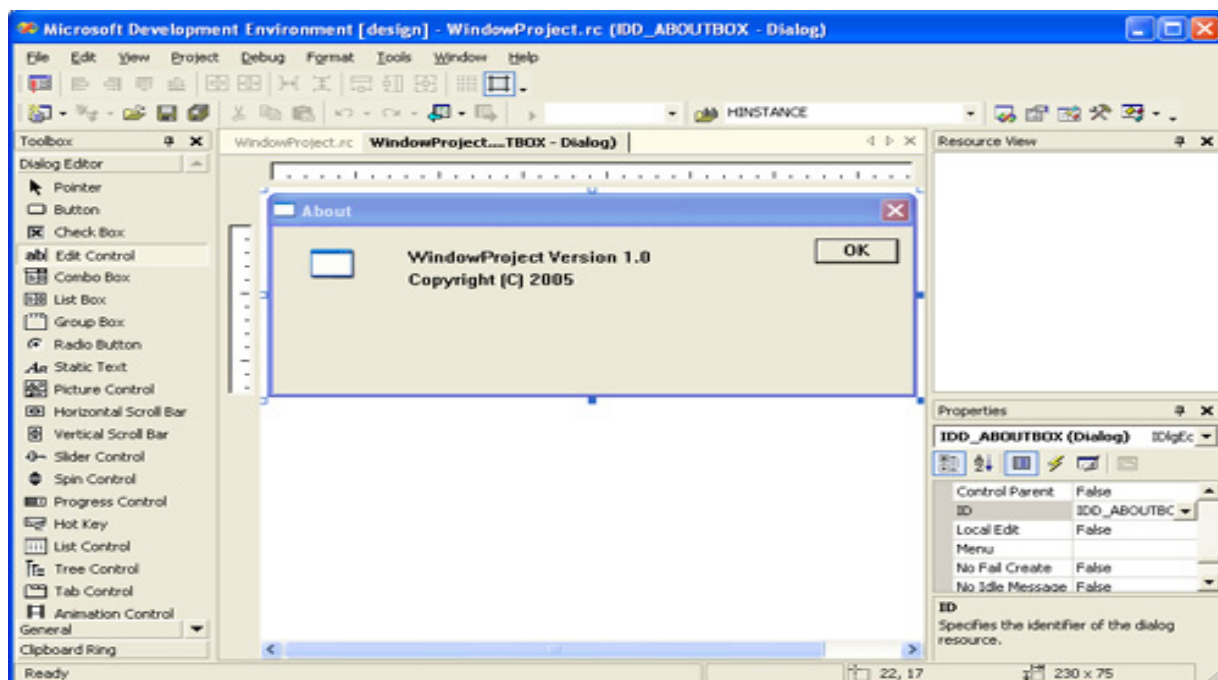
Для иконок принято в начале имени идентификатора добавлять префикс `IDI_`, что является сокращением от `Identifier Icon` (идентификатор иконки). Вы можете называть идентификаторы как угодно, но лучше все же придерживаться этого но лучше следовать этому негласному правилу.

Окна диалога

Наиболее интересным являются окна диалога, которые вы можете создавать визуально. Да, визуальные возможности редактора ресурсов сильно ограничены в количестве доступных элементов управления по сравнению с некоторыми другими визуальными языками программирования (например, по сравнению с Delphi), но все же, эти возможности удобны.

Данный документ распространяется только на диск к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

В нашем окне ресурсов в папке **Dialog** есть одно окно диалога с именем **IDD_ABOUTBOX**. Выделив его, в окне свойств вы сможете изменить это имя и идентификатор окна диалога. Дважды щелкните по имени диалога в дереве ресурсов, и среда разработки изменит внешний вид на визуальный редактор:



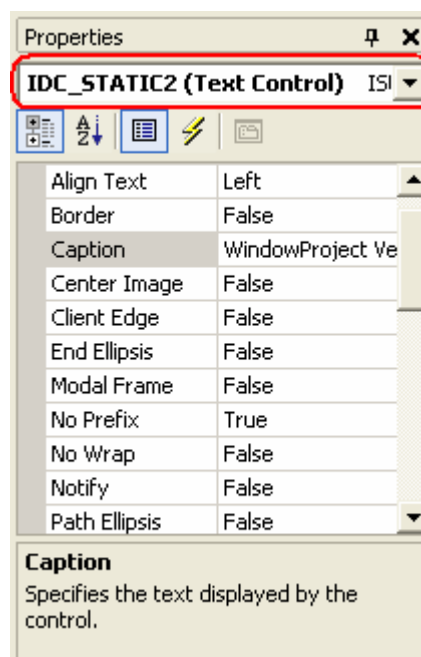
С левой стороны окна появилась панель, в которой можно увидеть кнопки, с помощью которых можно создать различные элементы управления. В центре окна расположен макет будущего окна диалога, и вы можете наблюдать, как он будет выглядеть визуально. С правой стороны остались окна **Resource View** и **Properties**. Выбирая на макете диалога различные элементы управления, в окне свойств появляются свойства этого элемента управления.

Давайте посмотрим, как можно редактировать это окно и создавать новые элементы управления.

На данный момент у нас на форме 4-е элемента управления:

- ☐ Picture Control – содержит иконку, которая отображается в окне слева;
- ☐ Text Control – содержит текст «WindowProject Version 1.0»;
- ☐ Text Control – содержит текст «Copyright (C) 2005»;
- ☐ Button Control – кнопка OK.

Выделять элементы управления, можно щелкая по ним на визуальной форме или выбирая их имена в выпадающем списке, который находится вверху окна **Properties**:



Чтобы выделять элементы с помощью мыши, необходимо выбрать в левой панели кнопку **Pointer**.

Static Text

Выделите первый текстовый элемент управления и посмотрите на его свойства. Их достаточно много, поэтому, чтобы проще было разобраться в их предназначении, обращайтесь вниз окна **Properties**. Посмотрите на рисунок выше, где выделено свойство **Caption**, а внизу окна появилась подсказка:

Caption

Specifies the text displayed by the control

На нашем великом и могучем это будет звучать следующим образом:

Заголовок

Определяет текст, отображаемый элементом управления



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Давайте изменим этот текст. Напишите в свойстве Caption какое-нибудь другое сообщение.

Давайте посмотрим, какие ещё свойства есть у элемента управления. Наиболее интересные это:

- ☐ **Sunken** (впалый) – если здесь установить true, то элемент управления будет выглядеть как бы утопленным;
- ☐ **Transparent** (прозрачный) – если установить в true, то отображаться будет только текст, а фон станет прозрачным и сквозь него можно будет увидеть элементы, которые находятся ниже. Например, это свойство будет удобно, когда текст находится поверх картинки, чтобы спрятать фон текста;
- ☐ **Word Ellipsis** – если текст не помещается в элемент управления, то в конце текста появиться три точки;
- ☐ **Align** – позволяет определить, по какому краю необходимо выравнивать текст. Это свойство имеет смысл, если элемент управления больше, чем нужно для отображения текста. Здесь можно выбрать одно из значений:
 - **Left** – выравнивать влево;
 - **Center** – выравнивать по центру;
 - **Right** – выравнивать вправо.
- ☐ **Border** – если в этом параметре указать true, то элемент управления будет окружен небольшой обложкой;
- ☐ **Client Edge** – если true, то нужно отображать клиентскую кромку.
- ☐ **Modal frame** – если true, то обложка будет, как у модального окна. Она выглядит, как будто текст выпуклый;
- ☐ **No wrap** – по умолчанию, если текст не помещается в элементе управления по ширине, то не поместившиеся слова будут перенесены на следующую строку. Если этот параметр равен true, то перевода не будет, текст будет в одну строку, и не поместившиеся слова просто исчезнут;
- ☐ **Visible** – элемент управления видим. Если этот параметр false, то вы его увидите только на форме дизайнера, а во время реального выполнения программы, элемент управления будет невидим.
- ☐ **ID** – идентификатор, а точнее имя. Числовой идентификатор создается автоматически, а вы можете только изменять имя константы;
- ☐ **Tab Stop** – если этот параметр true, то элемент управления должен получать фокус, когда пользователь нажимает клавишу tab. Для текстовых полей это не имеет смысла, поэтому лучше оставить значение false.

Поиграйте с параметрами, чтобы самостоятельно увидеть все возможные варианты отображения текста.

Теперь посмотрим, как можно создать новый текстовый элемент. Для этого в левой панели щелкните по кнопке **Static Text**, теперь щелкаем на окне диалога в том месте, где необходимо установить текст. При этом, будет создан элемент управления

со значениями по умолчанию. Если вы хотите изменить его размеры, то выделите, и растягивайте за квадратики, которые появятся вокруг элементы. Можно задать нужный размер элемента уже на этапе создания. Для этого нажмите левой кнопкой мыши на визуальной форме диалога, и, не отпуская мышки, растягивайте появившийся прямоугольник до нужных размеров.

Picture Control

Теперь посмотрим, что нам дает элемент управления **Picture Control**, который расположен в окне диалога слева. Выделите его и посмотрите на окно свойств. Наиболее интересные свойства:

- ☐ **Real Size Image** – если этот параметр равен true, то элемент управления Picture Control примет те же размеры, что и у картинки;
- ☐ **Image** – в этом выпадающем списке можно указать идентификатор картинки, которая должна отображаться. Эта картинка должна находиться в этом файле ресурсов. Если вы создали свою иконку, то попробуйте выбрать её в выпадающем списке этого свойства;
- ☐ **Type** – тип картинки. В нашем случае это иконка, но может быть и картинка Bitmap, просто окрашенный прямоугольник, и т.д.;
- ☐ **Center Image** – центрировать картинку. Если этот параметр равен true, то изображение будет находиться по центру элемента управления.

Мы рассмотрели только новые свойства, которые нам не встречались у элемента управления **Static Text**. Действительно, некоторые свойства одинаковы и имеют одно и то же значение. Опять же, поиграйте с параметрами, устанавливая разные значения, и посмотрите на результат.

Button

С кнопкой все ещё проще. Большинство из ее свойства нам уже знакомо. Среди новых свойств можно выделить:

- ☐ **Flat** – если параметр равен true, то кнопка будет плоской (плавающей);
- ☐ **Horizontal Alignment** – горизонтальное выравнивание текста на поверхности кнопки;
- ☐ **Multiline** – если параметр равен true, то текст на кнопке может быть многострочным;
- ☐ **Default Button** – если параметр равен true, то при нажатии клавиши Enter во время отображения диалога будет считаться, что пользователь нажал эту кнопку, вне зависимости от того, где сейчас фокус ввода.

Диалоговое окно

Само диалоговое окно также обладает свойствами. Щелкните в любом месте окна на дизайнера вне элементов управления и в окне Properties



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

появятся свойства диалогового окна. Здесь можно выделить следующие свойства:

- ☐ Set Foreground – если true, то данное диалоговое окно должно отображаться поверх всех окон, даже если главное окно программы не является сейчас активным;
- ☐ System Modal – если true, то окно будет модальным. Такие окна блокируют выполнение основного окна, пока модальное окно не будет закрыто;
- ☐ Font – шрифт по умолчанию, который будет использоваться диалоговым окном для отображения информации;
- ☐ Use System Font – если этот параметр равен true, то при отображении будет использоваться системный шрифт;
- ☐ Menu – здесь можно указать идентификатор меню, которое необходимо отобразить вверху окна диалога;
- ☐ Center – если этот параметр равен true, то окно будет отцентрировано по отношению к главному окну. Рекомендую выбирать этот параметр, чтобы приложение выглядело более привлекательным;
- ☐ X Pos – позиция по горизонтали, где необходимо отобразить окно;
- ☐ Y Pos – позиция по вертикали, где необходимо отобразить окно.

Check Box

На нашей форме нет элемента управления по имени Check Box, но это не значит, что он не пригодится вам в будущем. Этот элемент управления выглядит на экране следующим образом:



Он позволяет создавать элементы выбора, которые имеют два состояния true и false. У этого элемента управления можно выделить следующие свойства, значения которых нам ещё не известны, потому что они уникальны для данного элемента управления:

- ☐ Push Like – если этот параметр равен true, то элемент управления будет выглядеть как кнопка, которая может находиться в двух состояниях – нажата и нет;
- ☐ Tri-state – если этот параметр включен, то кнопка может находиться в трех состояниях – нажата, не нажата и не активна, иначе будут доступны первые два состояния;
- ☐ Auto – автоматически изменять состояние, когда пользователь щелкает на элементе управления. Если этот параметр равен false, то сколько не щелкай, автоматически состояние не измениться.

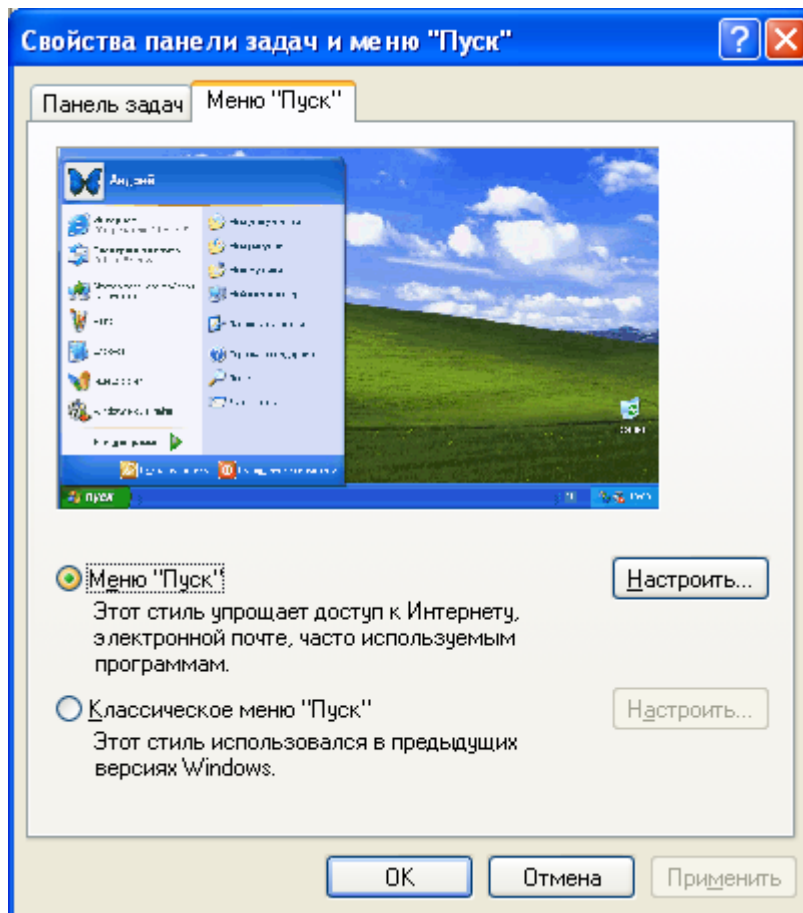
Radio Button

Элемент управления Radio Button позволяет делать выбор из нескольких вариантов и выглядит следующим образом:

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.



В следующем окне показан пример использования данного компонента для выбора между классическим видом меню кнопки Пуск и меню в стиле XP:



У этого элемента управления свойства очень похожи на элемент управления Check Box, поэтому их нет смысла рассматривать.

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Edit Control

Элемент управления Edit Control предназначен для ввода информации от пользователя и выглядит следующим образом:



Пояснения требуют следующие свойства элемента управления:

- ☐ Multyline – если этот параметр равен true, то будет доступен многострочный ввод;
- ☐ No hide selection – если true, то выделенный текст не будет прятаться, когда элемент управления потеряет фокус;
- ☐ OEM Convert – когда вы запросите текст из элемента управления, то он будет получен в формате OEM;
- ☐ Read Only – текст в элементе управления только для чтения. Его нельзя изменять;
- ☐ Auto HScroll – если true во время ввода информации будет автоматически происходить прокрутка по горизонтали;
- ☐ Auto VScroll – если true во время ввода информации будет автоматически происходить прокрутка по вертикали;
- ☐ Horizontal Scroll – если true, то элемент будет иметь горизонтальную полосу прокрутки;
- ☐ Vertical Scroll – если true, то элемент будет иметь вертикальную полосу прокрутки;
- ☐ Number – если true, то можно вводить только числа;
- ☐ Left Scrollbar – указывает на то, что вертикальная полоса прокрутки будет слева;
- ☐ Lowercase – вводимый текст будет приводиться к нижнему регистру;
- ☐ Uppercase – вводимый текст будет приводиться к верхнему регистру;
- ☐ Want return – если вводимый текст не будет помещаться по ширине, то текст будет переноситься на новую строку.

Остальные свойства нам уже знакомы или просто не заслуживают особого внимания, потому что используются слишком редко. А вот сам компонент используется очень часто, потому что трудно представить себе программу, на работу которой не может повлиять пользователь. Ввод информации нужен почти в любом приложении любого типа.



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Combo Box

Выпадающий список, позволяющий выбрать одно значение из списка и выглядит он следующим образом:

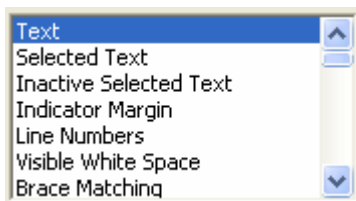


Из новых свойств, которые мы ещё не видели у уже рассмотренных элементов управления можно выделить два:

- ☐ Type – тип выпадающего списка. Наиболее интересными являются DropDown (выпадающий список с возможностью ручного ввода данных) и Drop list (можно только выбирать уже существующий элемент из списка);
- ☐ Sort – элементы в списке должны сортироваться.

List Box

Это тоже список, но не выпадающий. Здесь вы сразу видите все элементы и можете выбирать нужный. Иногда это удобнее, но отнимает на экране много места, потому что этот список не умеет сворачиваться. Выглядит компонент следующим образом:



У этого элемента управления можно выделить следующие свойства:

- ☐ Multicolumn – список может состоять из нескольких колонок;
- ☐ Selection – если этот параметр равен Single, то можно выделять только один элемент, а если Multiply, то можно выделять сразу несколько элементов списка.

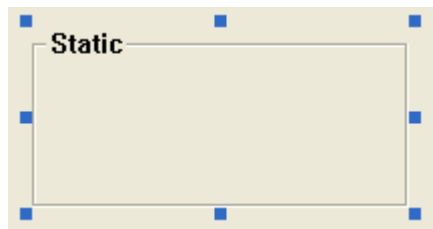


Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Group Box

С помощью этого элемента управления можно группировать другие элементы, располагая их поверх Group Box. Например, если расположить поверх группы 3 элемента Radio Button, то только один из них может быть выделен, а выделение с остальным будет автоматически сниматься. Но если на форме есть ещё элементы управления Radio Button, то на них влияния не произойдет, потому что они находятся вне группы.

Элемент управления Group Box выглядит следующим образом:



Новых свойств, заслуживающих внимания тут нет, потому что в основном вы можете настроить внешний вид элемента или изменить текст заголовка группы.

Horizontal Scroll Box и Vertical Scroll Box

Горизонтальная полоса прокрутки (Horizontal Scroll Box), которая выглядит следующим образом:

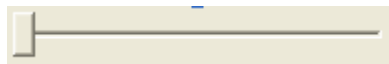


Вертикальная полоса (Horizontal Scroll Box) обладает теми же свойствами и таким же внешним видом, только вытянута по вертикали.

Тут можно выделить свойство Align, которое определяет, по какой стороне окна нужно выровнять элемент прокрутки. Больше ничего нового нет.

Slider control

Это ползунок, перетаскивая который можно выбрать определенное значение. Выглядит он следующим образом:



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Тут можно выделить следующие свойства:

1. Auto Ticks – автоматически расставить засечки на ползунке. Имеет смысл, если параметр Tick Marks равен true;
- ☐ Tick Marks – если true, то необходимо отобразить засечки;
 - ☐ Enable Selection – если true, то ползунок отображает только выделенный регион;
 - ☐ Orientation – горизонтальная или вертикальная ориентация;
 - ☐ Pont – где должны находиться засечки. Возможные варианты:
 - Both – с обеих сторон;
 - Top/Left – сверху или слева (зависит от ориентации);
 - Bottom/Right – снизу или справа (зависит от ориентации);

Spin Control

Компонент, который состоит из двух кнопок – одна со стрелкой вверх, а другая со стрелкой вниз:



Направленность кнопок можно изменить с помощью свойства Orientation. Такой компонент чаще всего используется совместно с полем ввода текста, когда вводимое значение должно быть числом. В этом случае, с помощью стрелок можно увеличивать или уменьшать число на единицу.

Progress Control

Компонент, который используется для отображения хода выполнения задачи и выглядит следующим образом:



Все свойства, которые доступны в компоненте, вам уже знакомы. Единственное – за ориентацию отвечает свойство Vertical. Если оно равно true, то бегунок движется вертикально, а по умолчанию горизонтально.

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

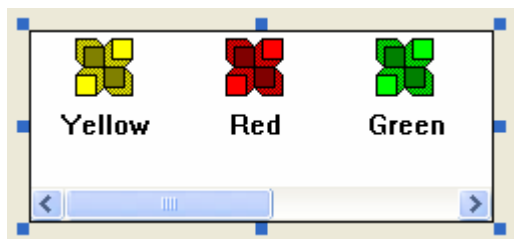
Hot Key

Удобный компонент для ввода горячих клавиш. Он похож на простой компонент Edit Control, но когда вы нажимаете клавиши, то в строке ввода появляются названия клавиш или сочетаний, которые вы нажали, например, на следующем рисунке показано, что нажато сочетание клавиш Ctrl + R:



List Control

Элемент списка, который позволяет отображать текст с иконками, например:



Такие компоненты очень часто используются для отображения файлов и папок, например, в проводнике Windows. У этого элемента есть некоторые свойства, которые нам ещё не знакомы и их стоит рассмотреть:

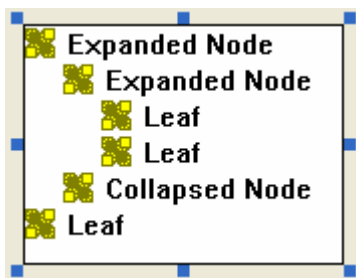
- ☐ Always Show Selection – если true, то выделенные элементы отображаются всегда, даже если элемент потерял фокус;
- ☐ Edit labels – если true, то заголовки можно редактировать;
- ☐ No column header – если true, то при отображении списка в виде отчета не показывать заголовки колонок;
- ☐ No Label Wrap – если true, то не переносить слова в надписях к элементам списка;
- ☐ No Scroll – если true, то прокрутка запрещена и считается, что все элементы видны на экране;
- ☐ Share Image List – если этот параметр равен true, то список картинок, который использовался для отображения не будет уничтожен при уничтожении элемента управления. Это удобно, если картинки используются и в других элементах управления;
- ☐ Single selection – если true, то только один элемент списка может быть выделен;
- ☐ View – параметры отображения. Здесь можно установить одно из значений:

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

- Icon – содержимое будет отображаться в виде больших иконок;
- Small icon – отображать маленькие иконки;
- List – отображать маленькие иконки в виде списка;
- Report – отображать в виде отчета.

Tree view

Дерево Tree View позволяет отображать элементы в виде дерева, например, как показано на следующем рисунке:



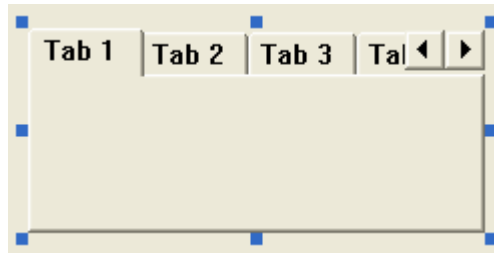
Тут у нас есть следующие интересные свойства:

- ☐ Check Boxes – если параметр равен true, то слева от каждого элемента управления будет отображаться квадратик Check Box, в котором можно ставить галочку для пометки элемента дерева;
- ☐ Disable Drag Drop – отключить возможность перетаскивания элементов;
- ☐ Full row select – выделять всю строку;
- ☐ Has buttons – если true, то папки будут иметь слева от имени квадратик с крестиком, по клику на котором можно открывать папку;
- ☐ Has lines – если true, то будут отображаться пунктирные линии, которые показывают связь между элементами;
- ☐ Lines at root – если true, то будут показываться линии, связывающие корневые элементы дерева;
- ☐ Single Expand – только одна папка будет открыта, а все остальные папки этого уровня дерева будут автоматически закрываться.

Tab Control

Компонент в виде закладок, на каждой из которых могут быть свои элементы управления. Этот компонент выглядит следующим образом:

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.



Animation Control

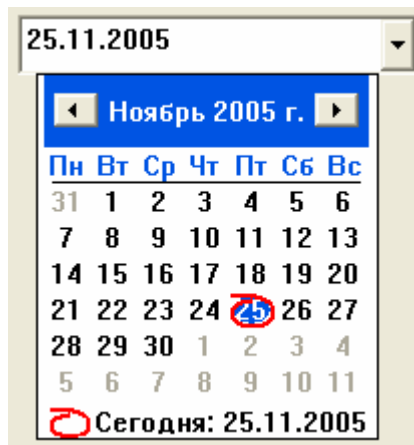
Компонент в котором можно отобразить анимацию в виде AVI файла или стандартную анимацию, такую как копирование файла, поиск, удаление и т.д.

Rich Edit

Многострочное поле ввода, которое позволяет выводить форматированный текст, наподобие того, что есть в программе Word Pad в составе Windows. Конечно, возможности форматирования не слишком велики, но они достаточны для решения простых задач;

Date Time Picker

Компонент для ввода дат, который имеет кнопку для вызова календаря. Этот элемент управления выглядит следующим образом:



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Тут можно выделить следующие свойства:

- ☐ Format – формат даты. В этом поле можно указывать следующие значения:
 - Short date – короткая дата;
 - Long date – длинная дата, при которой месяц отображается словом, а не цифрами;
 - Time – отображать время.
- ☐ Use spin control – если true, то вместо выпадающего календаря будет отображаться компонент Spin Control, с помощью которого можно увеличивать или уменьшать каждую составляющую дату или времени;
- ☐ Show none – если true, то будет отображаться элемент Check Box.

Этот элемент управления удобен тем, что он автоматически проверяет корректность даты. Вы просто не сможете ввести не корректную дату.

Month Calendar Control

Компонент позволяет отобразить красивый календарь, такой же, как выпадает у элемента управления Date Time Picker. Здесь можно выделить следующие свойства:

- ☐ No today – если параметр равен true, то внизу календаря не будет отображаться текущая дата;
- ☐ No today circle – если равен true, то не будет отображаться круг, по щелчку в котором будет происходить быстрый переход на текущую дату;
- ☐ Week numbers – если параметр равен true, то слева будут отображаться номера недель.

IP Address Control

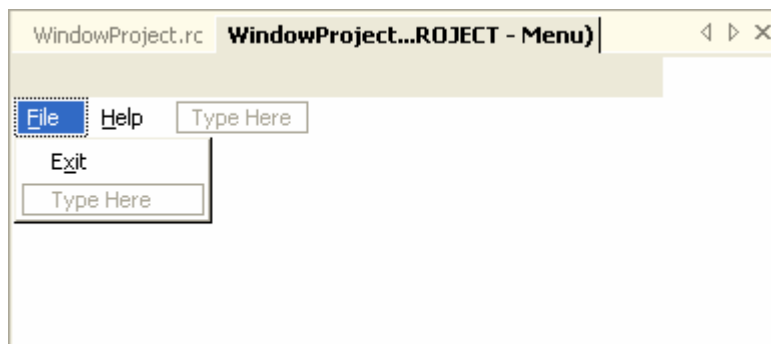
Поле для ввода IP адреса. Это поле специально заточено для этих нужд и состоит из четырех чисел разделенных точкой. Выглядит все это следующим образом:



Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Меню

Теперь поговорим о том, как создаются меню. Для этого в Visual Studio встроен удобный визуальный редактор меню. Раскройте папку **Menu** в окне **Resource View** и дважды щелкните по единственному существующему пункту меню



Обратите внимание, что в конце каждого раздела существуют пункты с названием **Type here**. Реально таких пунктов в меню не существует, это заготовки, которые можно выделить и можно набирать на них имя нового пункта меню. Давайте добавим раздел **Edit**. Для этого щелкните на пункте **Type here** справа от пункта меню **Help**, чтобы выделить его. Начните набирать слово «Edit». Обратите внимание, что ниже и справа от созданного пункта тоже появились пункты **Type here**, которые позволят расширять меню вправо и вниз. Теперь просто перетащите мышкой раздел меню **Edit** левее **Help**, где ему и место.

В данном случае мы вставляли новый раздел в самый конец меню, а потом перетаскивали его, а можно было сразу вставить его на своем месте. Давайте добавим в раздел **File** пункт меню **New**. Для этого выделите пункт меню **Exit** и нажмите кнопку **Insert**. Перед пунктом **Exit** появиться пустой пункт, где можно ввести имя **New**. Попробуйте сделать это.

Теперь попробуем создать разделитель – горизонтальная полоска, которая чаще всего ставится для отделения различных типов пунктов меню. Давайте добавим разделитель между пунктами **New** и **Exit**. Выделите пункт меню **Exit** и нажмите клавишу **Insert**, чтобы добавить новый пункт меню. Теперь, вместо ввода имени просто нажмите клавишу **тире**. Среда разработки добавит разделитель.

Таким образом, мы уже разобрались, как можно создавать меню, теперь давайте посмотрим, какие есть параметры у пунктов меню, которые мы можем изменять. Выделите какой-нибудь пункт и посмотрите в окно **Properties**. Наиболее интересные свойства, которые вы можете увидеть:

- ☐ **Caption** – заголовок пункта меню;
- ☐ **Checked** – если здесь **true**, то слева от имени будет галочка, т.е. пункт меню выбран;

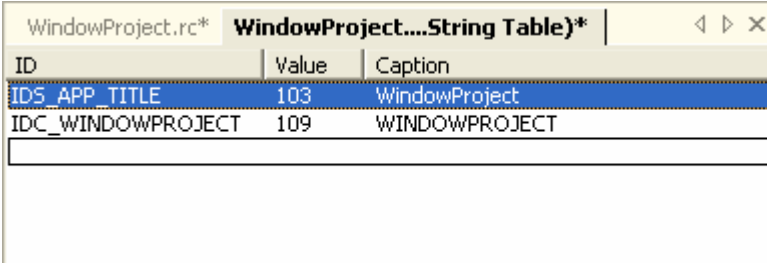
Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

- ☐ Enabled – пункт меню доступен. Если здесь установить false, то пункт будет не доступен, текст будет отображаться серым цветом и этот пункт нельзя будет выбрать;
- ☐ Grayed – если true, то пункт меню будет отображаться серым цветом;
- ☐ PopUp – указывает на то, что этот пункт может содержать элементы меню или подменю. Это свойство установлено у всех разделов и в большинстве случаев устанавливается автоматически. Вручную его нужно изменять только если вам необходимо программно добавлять подменю;
- ☐ Break – данный пункт является
- ☐ Separator – если это свойство равно true, то текущий оператор является разделителем.

Я настоятельно рекомендую изменять для пунктов меню их имена идентификаторов, потому что вы их будете связывать с обработчиком событий.

Таблицы строк

Теперь познакомимся с таблицами строк, которые вы можете увидеть в разделе **String Table**. Дважды щелкните по единственной существующей таблице строк, и перед вами откроется редактор, который имеет примерно следующий вид:



ID	Value	Caption
IDS_APP_TITLE	103	WindowProject
IDC_WINDOWPROJECT	109	WINDOWPROJECT

В данном случае у нас всего две строки, которые находятся в таблице из трех колонок:

- ☐ ID – имя идентификатора. По этому имени мы будем обращаться к строке;
- ☐ Value – числовой идентификатор;
- ☐ Caption – текстовая строка.

Во всех проектах, которые создает мастер, создается строка с идентификатором **IDS_APP_TITLE**, где храниться текст, который должен отображаться в заголовке окна. Чаще всего, этот текст далек от идеала и его следует изменить. Давайте посмотрим, как это сделать. Дважды щелкните по строке, которую нужно изменить, но только в колонке **Caption** и там появиться мини поле для ввода нового значения. Напишите новый текст заголовка окна. Можно просто выделить строку и изменить нужное свойство в окне **Properties**.

Данный документ распространяется только на диске к книге Фленова Михаила. Незаконное распространение через Интернет запрещено. Надеюсь, что вы будете следовать этому запрету.

Для создания новой строки щелкните по последней строке в таблице (она пустая) и заполните её значениями. Поле **Value** заполняется автоматически, первым свободным значением. Старайтесь изменять его аккуратно, чтобы не вызвать конфликта значений. Две строки не должны иметь одинакового значения.

Заключение

Мы рассмотрели основные типы ресурсов и как с ними работать. Остальные ресурсы, которые можно добавить и хранить в исполняемом файле, редактировать ещё проще. Например, картинки Bitmap редактируются также как и Icon (появляется простой текстовый редактор), а при создании HTML ресурса появляется простейший редактор HTML файлов. Надеюсь, что с этими ресурсами у вас не возникнет проблем.