

# **Vijeo Look.**

## **Краткое руководство пользователя,**

### **часть 4**

<< >>

<b>Введение в VBA</b>	<b>2</b>
VBA редактор	2
Объекты, свойства, методы и события	3
Остановка VBA программы	4
<b>Как VBA взаимодействует с HMI</b>	<b>5</b>
Проводник проекта VBA	5
Использование первичных графических элементов в программе	6
Использование RTDS переменных в программе	6
Первичные HMI объекты	8
<b>Примеры использования VBA</b>	<b>11</b>
Открытие графической страницы при запуске Vijeo Look	11
Получение и отображение имени текущего пользователя	12
Использование первичных графических элементов	13
Использование средств управления ActiveX	15

## Введение в VBA

Visual Basic for Applications (VBA)-это мощный инструмент разработки, базирующийся на Microsoft Visual Basic, разработанный для интеграции в приложения ПК, позволяющий ускорить их настройку и интеграцию с другими приложениями.

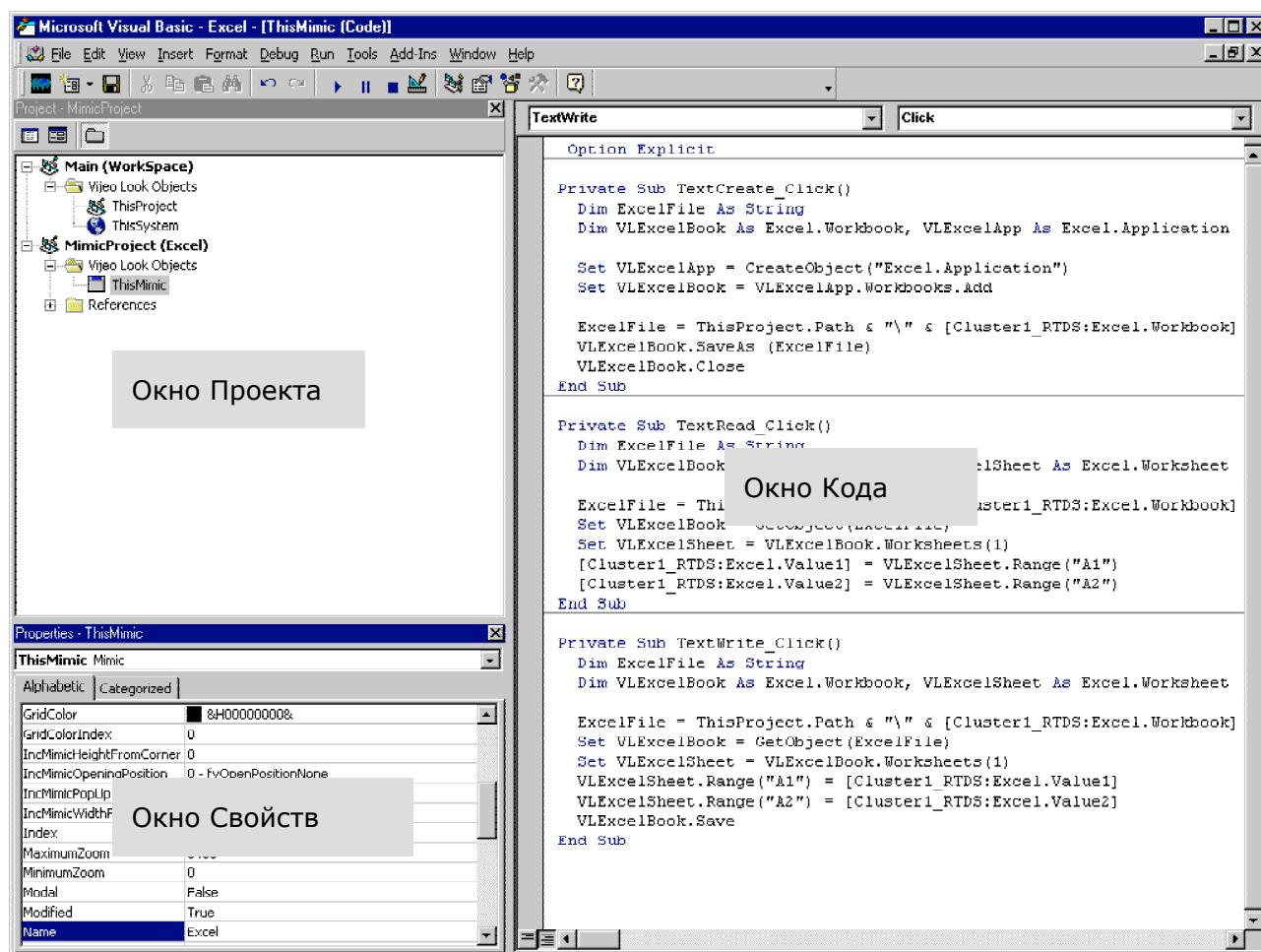
VBA –это объектно-ориентированный язык. В отличие от процедурных языков в нём используются объекты для создания приложений. Каждый тип объекта определен набором правил, называемых классами. Говорят, что объект - это экземпляр класса. Определения класса хранятся в библиотеках типов и могут быть просмотрены программой просмотра объектов. Объекты и классы упрощают кодирование и облегчают повторное использование кода.

Разработка кода ведётся с использованием редактора VBA, встроенного в Vijeo Look.

### VBA редактор

VBA редактор встроен в Vijeo Look и может быть открыт из стандартной панели инструментов или с помощью клавиатуры с использованием клавиш Alt F11.

Ниже перечислены основные элементы VBA редактора. (VBA редактор-это очень богатая среда и ее полное описание не входит в рамки данного описания. Подробную информацию смотрите в соответствующей документации.)



### Окна

Есть ряд окон, которые могут быть открыты в редакторе. Каждое - имеет свою функцию. Следующие окна наиболее часто используются и отображаются по умолчанию.





Окно Кода - используется для написания, отображения и редактирования кода Visual Basic. Можно открыть столько окон Кода, сколько модулей вы имеете, а также вы можете легко просмотреть код в различных формах или модулях, копировать фрагменты кода и вставлять его в различные места модулей.

В Окне Свойств перечисляются проектируемые свойства для выбранных объектов и их текущие параметры настройки. Вы можете изменить эти свойства во время проектирования.

Окно Проекта отображает иерархический список проектов и всех элементов, содержащихся или адресуемых в каждом из проектов.

### Панели инструментов

Имеется четыре панели инструментов: Отладка, Редактирование, Стандартная и Форма Пользователя (UserForm). Из них Стандартная панель инструментов отображается по умолчанию, и будет использоваться наиболее часто. Обычно применяются следующие инструменты.

Инструмент	Использование
	Отобразить Vijeo Look HMI.
	Сохранить текущий модуль.
	Переключить режимы Исполнения и Проектирования.
	Запустить подпрограмму или Форму Пользователя.

### Объекты, свойства, методы и события

Объекты - это фундаментальный блок построения VBA; почти все, что вы делаете в VBA, влечёт модификацию объектов. Каждый элемент HMI, окна, рисунки, анимации, средства управления ActiveX и т.д. могут быть представлены объектом на языке скриптов.

#### Что такое свойства?

Свойство - это атрибут объекта или аспект его поведения. Например, свойства графической страницы включают имя, размер и ее позицию. Чтобы поменять характеристики объекта, вы меняете значения его свойств.

Чтобы установить значение свойства, используйте ссылку на объект с точкой и именем свойства, знак равенства и новое значение свойства. В следующем примере изменяется цвет нарисованного элемента "Shape\_Red."

```
Shape_Red.BackColorPrimary = RGB(255, 0, 0)
```

Вы можете получить информацию об объекте с помощью возвращаемых значений одного из его свойств. В следующем примере возвращается высота нарисованного объекта.

```
ObjHeight = Object_Truck.Height
```

В этом примере Object\_Truck ссылается на нарисованный элемент. Высота этого нарисованного элемента присваивается переменной ObjHeight.



Обратите внимание, что не все свойства могут быть установлены - некоторые только для чтения.

#### Что такое Метод?

Метод-это действие, которое объект может выполнить. Например, наряду с тем, что нарисованный элемент может быть перемещен, объект элемента рисунка имеет метод Перемещение. Методы обычно имеют параметры, которые определяют, как выполняется действие. В следующем примере перемещается точечный рисунок "Bitmap1" к новому местоположению в пределах его окна.

`Bitmap1.Move NewXPos, NewYPos`

В большинстве случаев методы – это действие, а свойства – качества. Использование метода производит какое-то действие с объектом, в то время как при использовании свойств возвращается какая-то информация об объекте, или изменяется качество объекта.

### **Что такое Событие?**

Событие- это действие, распознаваемое объектом, такое как нажатие кнопки мыши или нажатие клавиши, для которого вы можете написать код, чтобы отреагировать на него. События могут происходить по ряду причин:

- Действия пользователя - такие как нажатие кнопки мыши или нажатие клавиши.
- Изменение значения RTDS переменной.
- Из-за другого кода программы.
- Что-то произошло в системе.

### **Остановка VBA программы**

Рассмотрение этой темы может показаться излишним, но когда вы плохо знакомы с VBA, вы, скорее всего, захотите остановить VBA программу, которая не может быть завершена нормальным образом (например, если она находится в бесконечном цикле). Чтобы сделать это, нажмите Ctrl + Break, находясь в рабочей области Vijeo Look.

## Как VBA взаимодействует с НМІ

Эта тема охватывает специфику выполнения VBA в Vijeo Look.

### Проводник проекта VBA

В Проводнике проекта VBA доступны только Проекты, Элементы Проекта, Средства Управления и нет никаких других. Из-за этого ограничения одно НМІ приложение выглядит в Проводнике Проекта как несколько различных проектов.

#### Проект Main

Проект Main существует всегда и обеспечивает доступ к двум элементам проекта ThisProject и ThisSystem. Проект Main автоматически добавляется как ссылка ко всем другим проектам (выглядит как ссылка на Рабочую Область в проводнике проекта), так что вы можете обращаться к его функциям и переменным, как будто они были глобальными.

#### Проект MimicProject

Каждая открытая графическая страница появляется в отдельном MimicProject проекте. В MimicProject есть один элемент проекта ThisMimic, представляющий объект - графическую страницу. Каждый MimicProject имеет свои собственные режимы проектирования и исполнения. При переключении режимов, графическая страница и связанный с ней MimicProject следуют один за другим. В режиме проектирования события не обрабатываются.

#### Проект SymbolProject

Каждый символ, отображаемый в открытой графической странице, появится в отдельном проекте SymbolProject . Отображается только один SymbolProject для каждого типа символов, даже если он есть в нескольких позициях на графической странице. В SymbolProject есть один элемент проекта ThisSymbol, представляющий объект Символ.

#### Глобальные Модули

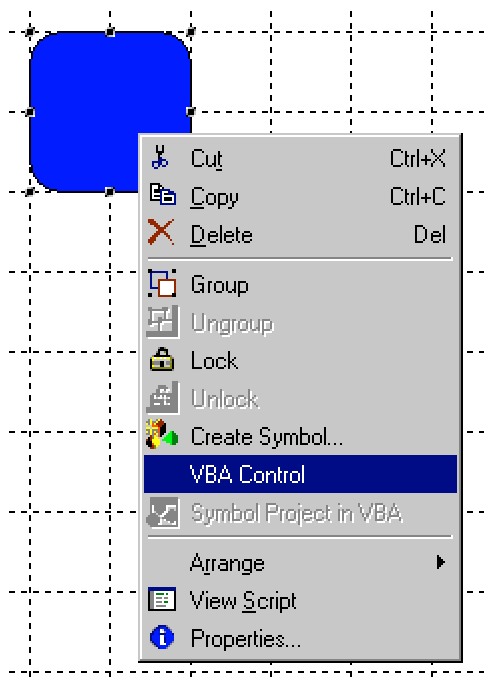
Вы можете вставлять Модули и Модули Класа в любом проекте VBA . Модули в проекте Main будут видны во всех других проектах. В случае конфликта имен используйте "Main." в качестве приставки.



ThisProject это элемент типа fvProject .Он эквивалентен типу Project, но переименован, чтобы избежать конфликта с объектом VBA Project.

## Использование первичных графических элементов в программе

Перед использованием любого простейшего графического элемента (Форма, Текст, Точечный рисунок и т. д.) как средства управления в VBA программе, вы должны выбрать свойство *VBA Control* во всплывающем меню, отображаемом при нажатии правой кнопки мыши на объекте.



Как только графический элемент становится средством управления VBA, он будет видимым VBA среде, и свойство *EnableEvents* графического элемента устанавливается в *True*.

- Он может быть выбран в левом поле со списком в окне кода.
- Выбрав в левом поле со списком, вы можете выбрать любое из его событий в правом поле со списком и непосредственно закодировать событие.
- Он виден из блока свойств VBA.
- Объект известен в проекте VBA непосредственно под своим именем (shape1 например).

Вы можете установить/сбросить свойство Средство Управления VBA в любое время. Связанный код не теряется: если вы сбросите Средство управления, он становится функцией проекта.



Для графических элементов возможно, чтобы он являлся средством управления VBA без установки свойства *EnableEvents* в *True*. В этом случае вы можете изменить его свойства программно, но не сможете запустить любое событие



Каждый графический элемент, установленный как средство управления, требует дополнительных ресурсов от ПК. Устанавливайте графические элементы в качестве средства управления только в том случае, если вам необходимо разработать для них код.

## Использование RTDS переменных в программе

Семейство *Variables* (также *TheseVariables*) обеспечивает доступ ко всем текущим описанным переменным. Переменные описываются автоматически при использовании в анимации в открытой графической странице, или могут быть описаны вручную, используя

метод `Variables.Add`. Набор содержит как RTDS, так и временные переменные (с именами, заканчивающимися знаком %).

### Как описать переменную?

Чтобы описать переменную, используйте метод `Variables.Add`. Например:

```
Variables.Add "Main:Branch1.B1", fvVariableTypeBit
Variables.Add "CommentDisplay%", fvVariableTypeText
```

Необязательный второй параметр описывает тип переменной. Для получения дополнительной информации смотрите раздел Семейство Переменных (`Variables Collection`).

Если переменная уже описана с помощью анимации, она еще может быть описана с использованием этого метода. Когда графическая страница, содержащая анимацию, закрыта, переменная остается описанной.



Вы должны использовать полное имя переменной, включающее имя кластера, если это OPC переменная.

### Как удалить переменную?

Чтобы удалить переменную, используйте метод `Variables.Remove`. Например:

```
Variables.remove "Main:Branch1.B1"
```

### Ярлыки для доступа к Переменным

Метод `Item` установлен по умолчанию для набора переменных, и свойство `Value` установлено по умолчанию для объекта `Variable`. Поэтому:

```
Variables("CommentDisplay") = "Hello World"
```

Это эквивалентно:

```
Variables.Item("CommentDisplay").Value = "Hello World"
```

Синтаксис[...] обеспечивает дополнительный ярлык для непосредственного обращения к переменной.

```
[Main.Branch1.B1]
```

Это эквивалентно:

```
Variables("Main.Branch1.B1")
```

Использование такого ярлыка делает ваш код более читаемым. Например, Вы можете написать:

```
Private Sub Shape1_Click()
    If [Main.Branch1.B1] = True Then
        [Main.Branch1.B1] = False
    Else
        [Main.Branch1.B1] = True
    End If
End Sub
```



Если код содержится внутри графической страницы, любые ссылки на объект временны. Когда графическая страница закрыта или изменен ее режим (с Исполнения на Проектирование), ссылка теряется, и событие больше не будет обрабатываться. Для того чтобы событие снова стало обрабатываемым, объект должен быть повторно обозначен переменной и разрешены события.

## Первичные HMI объекты

Свойства, методы и события HMI объектов позволяют Пользователю разрабатывать любые, требуемые ему алгоритмы для наблюдения и контроля за процессом. Некоторые из объектов всегда доступны как часть функционального ядра HMI (напр. ThisProject), в то время как другие могут быть созданы как часть приложения (например, открытая графическая страница).

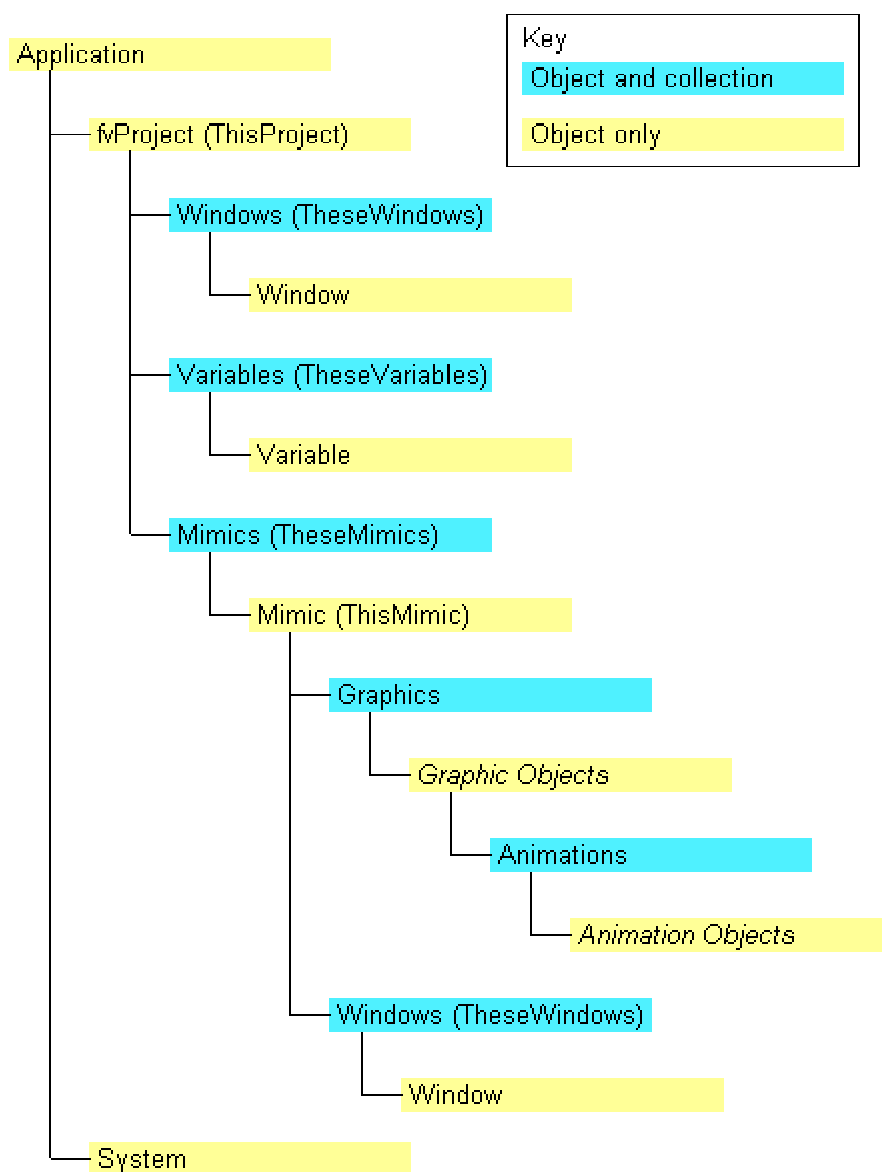
Класс	Представление	Элементы HMI
ActiveX	Объект ActiveX - посредник. Его свойства, методы, функции и события предъявляются один раз при его вставке в графическую страницу. Для информации о поведении объекта см. документацию, предоставляемую поставщиком.	
Application	HMI <i>приложение</i> .	Приложение
AVI	Объекты AVI.	
Bitmap	Объект - <i>точечный рисунок</i> . Этот класс поддерживает графические форматы BMP и JPG.	
fvProject	Текущий открытый проект.	ThisProject
GIF	Объект GIF . Этот класс поддерживает формат рисунков GIF.	
Graphics	Семейство, представляющее все нарисованные объекты на графической странице. Включает объекты следующих классов. ActiveX AVI Bitmap GIF Group Metafile OLE Poly Shape Symbol Text	Graphics
Group	Объект <i>группа</i> . Множество элементов рисунка, которые были сгруппированы вместе. (Но это не символ, вставленный из библиотек)	
Metafile	Объект metafile. Этот класс поддерживает форматы изображения WMF и EMF.	
Mimic	Открытая графическая страница.	ThisMimic
Mimics	Семейство, представляющее все открытые графические страницы.	TheseMimics, Mimics



Класс	Представление	Элементы HMI
OLE	OLE объект - посредник. Его свойства, методы, функции и события предъявляются один раз при его вставке в графическую страницу. Для информации о поведении объекта см. документацию, предоставляемую поставщиком	
Poly	Объект <i>многоугольник/ломаная</i> . Этот класс включает Линии, Ломаные, Многоугольники и Кривые Безье.	
Shape	Объект <i>форма</i> . Этот класс включает Прямоугольник, Закругленный Прямоугольник, Эллипс, Дугу и Часть Дуги.	
Symbol	Объект <i>символ</i> . Первичный HMI-символ из библиотеки символов. (Предварительно созданный как объект).	
System	Набор свойств, который описывает систему, используемую для исполнения HMI.	ThisSystem, System
TemplateGraphics	Семейство, представляющее все нарисованные объекты на шаблоне графической страницы. Идентично семейству Graphics за исключением того, что указывает на Graphics шаблона страницы. Если графическая страница не имеет шаблона, тогда TemplateGraphics перенаправляется в Graphics	TemplateGraphics
Text	Объект <i>текст</i> . Этот класс включает текстовые графические элементы	
Variable	Постоянно описанная переменная. Класс обеспечивает свойства, методы и события для переменных всех типов.	
Variables	Семейство, представляющее все постоянно описанные переменные.	Variables, TheseVariables
Window	Вид открытой графической страницы. Графическая страница может иметь несколько видов одновременно. Например, один в режиме проектирования, а другой в режиме исполнения.	
Windows	Семейство, представляющее все открытые объекты Window.	Windows

## Иерархия объектов НМІ

Для управления первичными НМІ объектами пользователь должен знать отношения между ними. Объекты самого верхнего уровня - это *Приложение* со всеми другими подчиненными ему объектами. При использовании методов и свойств объекта *Приложение* вы можете обратиться ко всем подчиненным ему объектам. Иерархия объектов показана ниже.



*Graphic Objects* –это родовой терм, объединяющий все остальные графические классы объектов (*Форма, Текст* и т. д.).

*Animation Objects* –это родовой терм, объединяющий все классы анимированных объектов. Специальные классы анимации не доступны в данной версии.

Имеется также семейство *TemplateGraphics*, которое является идентичным семейству *Graphics* за исключением того, что ссылается на *Graphics* в шаблоне графической страницы.

## Примеры использования VBA

Следующие примеры иллюстрируют несколько типовых случаев (если такая вещь вообще возможна) использования VBA в среде Vijeo Look .

### Открытие графической страницы при запуске Vijeo Look

Вы можете открыть одну графическую страницу автоматически при запуске Vijeo Look, используя опцию, доступную в дереве Options. Однако иногда при запуске необходимо открыть больше, чем одну графическую страницу, например Menu и страницу Status. Следующий пример показывает один из способов того, как сделать это.

#### Предварительная подготовка


1. Создайте новую графическую страницу и сохраните ее со следующими свойствами.

Width	1024
Height	86
Sizeable	Yes
Border	Yes
Name	Status

#### Разработка кода

1. Откройте редактор VBA (нажав Alt-F11 или с использованием соответствующего инструмента на стандартной панели инструментов).
2. Окно Проекта должно быть отображено в левой части редактора с единственным отображенным на нем элементом Main(Workspace). Двойным щелчком на нём раскройте дерево.
3. Теперь вы должны увидеть второй элемент - Объекты Vijeo Look . Еще раз щёлкните дважды, и вы увидите еще два элемента, представляющих объекты ThisProject и ThisSystem.
4. Дважды щёлкните по ThisProject. Откроется окно кода.
5. Сверху окна кода имеются два списка. Список слева должен показывать fvProject. В списке справа выберете StartupComplete. Теперь вы должны видеть начало и конец модуля кода для этого события.
6. Между строками Sub и End sub напечатайте следующую строку:

```
TheseMimics.Open ("Status")
```

7. Нажмите на значок Save , чтобы сохранить написанный код.




Выполняя это, вы должны заметить одну из главных особенностей VBA-редактора. Как только вы печатаете после имени объекта точку, вы автоматически выводите список доступных свойств или методов.




Объект TheseMimics тоже известен как семейство, поскольку он ссылается на все графические страницы. Open-это метод, который открывает указанную графическую страницу.

#### Проверка кода

На данном этапе изучения Vijeo Look вы не должны создавать события для того, чтобы посмотреть результирующее действие. Однако проверить код Вы можете следующим образом.

1. Убедитесь, что в VBA редакторе окно кода, содержащее только что написанный подмодуль, активно.
2. Запустите код, используя либо команду меню **Run.Run Sub/UserForm** ,либо нажав функциональную клавишу F5, либо с помощью значка .

3. Вернитесь в Vijeo Look HMI, нажав на иконку . Открывается графическая страница Status.


## Получение и отображение имени текущего пользователя

Полезно поместить имя текущего пользователя в текстовую переменную. Тогда оно может быть отображено на странице, и любые изменения пользователя зарегистрируются с помощью HDS

### Предварительная подготовка

1. Откройте Конфигуратор.
2. Создайте новый подкаталог(sub-branch) System в папке Variables.
3. Создайте новую текстовую переменную UserName в подкаталоге System.  
Убедитесь, что отметили свойство *HMI access*.
4. Закройте Конфигуратор.
5. Откройте страницу Status и выберите режим проектирования.
6. Нарисуйте текстовую строку. Анимлируйте ее, используя Text animation и только что созданную переменную (Cluster1\_RTDS:System.UserName).
7. Сохраните и закройте графическую страницу.

### Разработка кода

1. Откройте VBA-редактор и отобразите окно кода для ThisProject. (Дважды щёлкните по значку ThisProject в окне проекта.)
2. В программном модуле обработки события fvProject\_StartupComplete напечатайте следующую строку:  
`Variables.Add "Cluster1_RTDS:System.UserName", fvVariableTypeText`
3. Теперь обязательно внесите комментарий в ваш код. VBA интерпретирует любую строку, начинающуюся с одиночной кавычки, как комментарий. Выше кода, который вы только что напечатали, введите соответствующий комментарий. Например:  
`'----- Описание RTDS переменных`
4. В верхней части окна кода выберите событие UserChanged. Теперь Вы должны видеть начало и конец модуля кода для этого события.
5. В программном модуле обработки события fvProject\_UserChanged напечатайте следующую строку:  
`[Cluster1_RTDS:System.UserName] = ThisProject.UserName`
6. Сохраните написанный код.
7. Вернитесь в Vijeo Look HMI, нажав иконку .



При описании объекта Variable вы предъявляете его свойства для VBA.



`[Cluster1_RTDS:System.UserName]` эквивалентно:  
`Variables.Item("Cluster1_RTDS:System.UserName").Value`

### Проверка кода

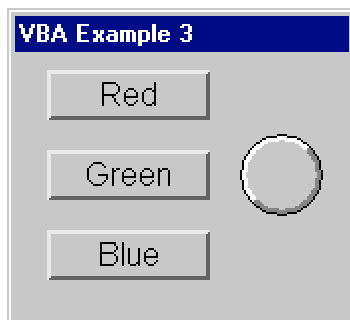
1. Откройте страницу Status. Анимация, отображающая переменную Cluster1\_RTDS:System.UserName, возможно, будет помечена знаком ? Это показывает, что она сейчас недоступна.
2. Зарегистрируйтесь, используя одно из заранее выбранных имен Пользователя. Последнее можно сделать с помощью клавиатуры, используя F2
3. Новое имя Пользователя появится на графической странице Status.

## Использование первичных графических элементов

Следующий пример показывает простой пример взаимодействия VBA с первичными графическими элементами HMI.

### Предварительная подготовка

1. Создайте новую страницу и сохраните ее под именем Buttons.
2. Нарисуйте три текстовых строки с текстом Red, Green и Blue. Задайте на закладке Aspekt внешний вид строк *Inverse Relief*.
3. Нарисуйте эллипс и выберите внешний вид *Coloured Button*. В режиме исполнения графическая страница должна выглядеть примерно так:



4. Выберите режим разработки для графической страницы и откройте список Properties, используя команду меню **Display.Properties List**.
5. Выберите текстовую строку Red. В списке Properties измените её имя Name на *TextRed*. Это имя, под которым графический элемент будет известен в среде VBA. (Выберите Properties в Properties List, если они ещё не выбраны.)
6. Нажмите правой кнопкой мыши на текстовую строку Red и выберите из всплывающего меню **VBA Control**. Это приведёт к появлению окна со свойствами и событиями графического элемента для VBA.
7. Повторите шаги 5 и 6 для двух других строк, используя имена *TextGreen* и *TextBlue*.
8. Повторите шаги 5 и 6 для эллипса, используя имя *ShapeInd1*.
9. Сохраните графическую страницу.

### О событиях для графических элементов

Для графических элементов HMI все события происходят в ответ на операции с мышью.

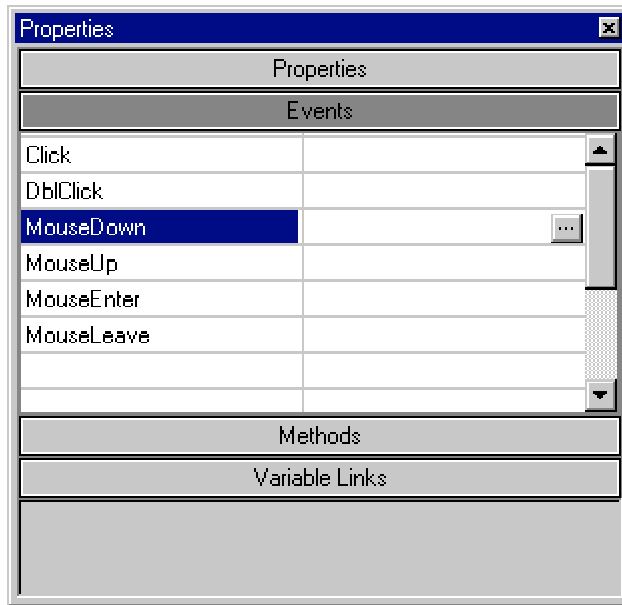
Название события	Описание
Click	Происходит при щелчке левой кнопкой мыши.
DoubleClick	Происходит при двойном щелчке левой кнопкой мыши.
MouseDown	Происходит, когда нажимается любая из кнопок мыши.
MouseUp	Происходит, когда отпускается любая из кнопок мыши.
MouseEnter	Происходит, когда указатель мыши входит в область, занятую графическим элементом.
MouseLeave	Происходит, когда указатель мыши выходит из области, занятой графическим элементом.

### Разработка кода

Первое, что надо сделать – это заставить графический текст вести себя как кнопки. Чтобы сделать это, мы можем менять внешний вид динамически, используя VBA.

1. Выберите режим разработки для графической страницы Buttons и откройте список свойств Properties, используя команду меню **Display.Properties List**.
2. Выберите текстовую строку Red. В списке Properties выберите Events. Отобразится список доступных значений.

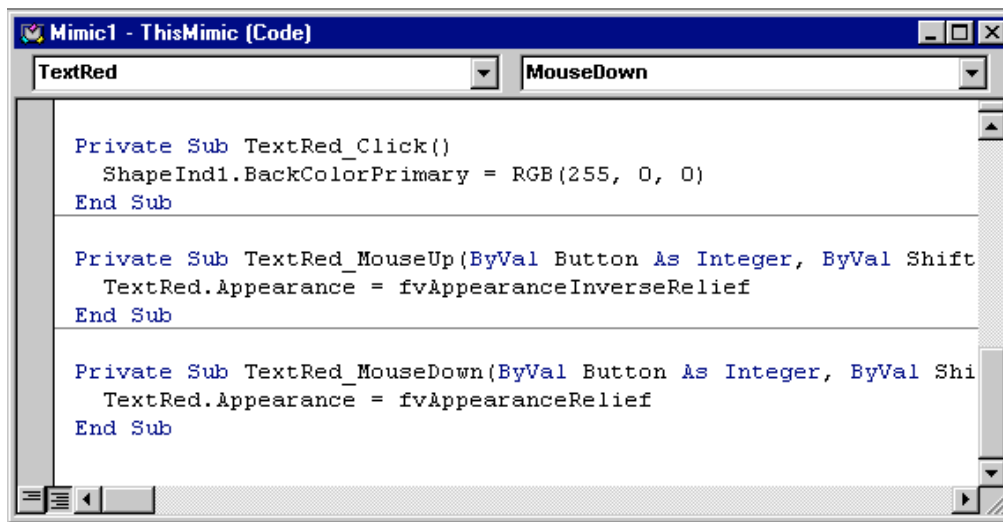
3. Выберите событие *MouseDown* и щелкните по кнопке эллипса, смежной с ним. Откроется VBA-редактор с автоматически созданной подпрограммой `TextRed_MouseDown`



4. В редакторе напишите текст `TextRed`, заканчивающийся точкой. В этот момент редактор автоматически отобразит список доступных свойств для объекта `TextRed`. Это стандартная особенность VBA-редактора - автоматически обеспечивать список свойств для известных типов объектов. Выберите свойство `Appearance` (Внешний Вид).
5. Продолжайте, напечатав знак `=`. Редактор автоматически отобразит список возможных значений для свойства `Appearance`. Выберите `fvAppearanceRelief`.
6. Возвратитесь в HMI и повторите шаги 3, 4 и 5 для события `MouseUp`, выбрав в этот раз значение `fvAppearanceInverseRelief`.
7. Сохраните написанный код и вернитесь в HMI. Выберите режим исполнения для графической страницы. Теперь, когда вы нажмете на строку `Red`, будет казаться, что объект поднимается и опускается как кнопка.
8. Повторите шаги 3, 4, 5 и 6 для текстовых строк `Green` и `Blue`.

Теперь мы имеем несколько текстовых строк, которые ведут себя как кнопки, но больше ничего не делают. Мы добавим код, который заставит цвет эллипса измениться.

1. Выберите режим разработки для графической страницы `Buttons` и откройте список `Properties`, используя команду меню **Display.Properties List**.
2. Выберите текстовую строку `Red`. В списке `Properties` выберите `Events`. Отобразится список доступных значений.
3. Выберите событие `Click` и щелкните по кнопке эллипса, смежной с ним. Откроется редактор VBA с автоматически созданной подпрограммой `TextRed_Click`.
4. Напечатайте следующий код `ShapeInd1.BackColorPrimary = RGB(255, 0, 0)`. Как и раньше, когда вы напечатаете имя известного объекта, сопровождаемое точкой, отобразится список доступных свойств объекта.
5. Повторите шаги 3 и 4 для текстовых строк `Green` и `Blue`, используя `RGB(0, 255, 0)` и `RGB(0, 0, 255)` соответственно.
6. Сохраните написанный код и вернитесь в HMI.



Функция RGB возвращает число, представляющее значение цвета RGB.

### Проверка кода

1. Выберите режим исполнения для графической страницы.
2. Нажмите на кнопку Red. Эллипс должен закраситься в Красный цвет.
3. Повторите эти действия для других кнопок и наблюдайте эффект.

## Использование средств управления ActiveX

Средство управления ActiveX - программные компоненты, основанные на общей архитектуре, которая позволяет взаимодействовать программному обеспечению под управлением Microsoft Windows. Чтобы использовать средство управления ActiveX, вы должны объединить его с другой программой, которая была разработана как контейнер ActiveX. Имеются много программ, являющихся контейнерами ActiveX, включающими HMI. Связь между средством управления ActiveX и программой, содержащей его, основана на использовании методов, событий и свойств.

Для этого примера мы будем использовать средство управление ActiveX по имени AITimer. Он предусмотрен HMI, и его функция состоит в том, чтобы создать события в определенном интервале.

### Предварительная подготовка

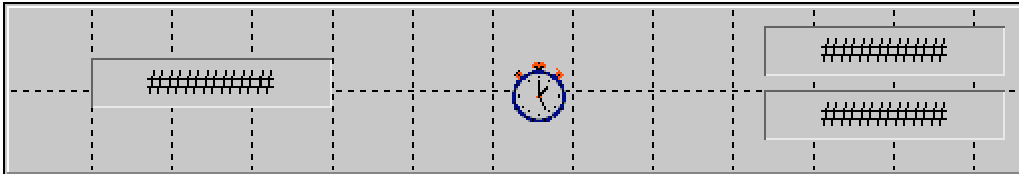
1. Создайте две новых текстовых переменных Time и Date в подкаталоге System. Удостоверьтесь, что вы отметили свойство *HMI access*.
2. Откройте графическую картинку Status и выберите режим разработки. Создайте два текстовых графических элемента и анимируйте их, чтобы отобразить две текстовые переменные, которые вы только что создали.
3. Сохраните графическую страницу.

### Авторизация и вставка средства управления ActiveX.

Прежде, чем средство управление ActiveX может быть использовано с HMI, оно должно быть авторизовано. В этом процессе проверяется соответствие интерфейса ActiveX стандартам, установленными Microsoft. (Если вы используете несовместимый ActiveX, это может разрушить HMI)

1. Из панели меню выберите строку **Tools.Prefences.ActiveX Controls**. В левой области окна - список средств управления ActiveX, известных Windows. В правой области окна - список средств управления, доступных HMI.

2. Выберите в левой области средство управления AITimer Control и нажмите на клавишу в виде двойной стрелки вправо. HMI проверит целостность средства управления и оно появится в правой области. Закройте окно ActiveX Control.
3. Откройте графическую страницу Status и выберите режим разработки.
4. Сформируйте меню, выбрав **Insert.ActiveX Control**. Откроется блок, отображающий список авторизованных средств управления ActiveX.
5. Выберите строчку AI Timer Control и нажмите кнопку OK . AI Timer Control появится на графической картинке, а список ActiveX Control закроется.
6. Отобразите список Properties, используя команду меню **Display.Properties List**.
7. Выберите AI Timer Control. Поменяйте свойство *Interval* на 1000 (миллисекунд). Поменяйте свойство *Enabled* на False.
8. Сохраните графическую страницу. Она может выглядеть примерно так:



### Разработка кода

1. Выберите ActiveX Timer и из списка Properties выберите событие Timer. Откроется VBA-редактор с подпрограммой обработки событий, называемой Timer1\_Timer.
2. Напечатайте следующий код.  

```
[Cluster1_RTDS:System.Time] = Time
[Cluster1_RTDS:System.Date] = Date
```
3. Сохраните код и вернитесь в HMI.



Time и Date –это функции, возвращающие Время и Дату.

### Проверка кода

Как только в свойстве Enable установлено значение true, средство управления ActiveX начинает генерировать события. Если в вашем VBA-коде имеются ошибки, вы можете попасть в ситуацию, где ошибки будут непрерывно генерироваться прежде, чем вы получите возможность исправить их. Поэтому лучше проверить код вручную прежде, чем разрешить использование событий в последующих процедурах.

1. Переключите графическую страницу Status в режим исполнения. Две анимации, показывающие Время и Дату, будут иметь предупреждения напротив них, показывающие, что переменная недоступна.
2. Откройте VBA-редактор и выберите подмодуль Timer1\_Timer.
3. Щелкните по кнопке Run sub/UserForm на панели инструментов. Этим будет вызвано однократное исполнение кода Timer1\_timer.
4. Вернитесь в HMI. Если все хорошо, анимации Date и Time должны содержать значения.
5. Переключите графическую страницу Status в режим проектирования.
6. Выберите средство управления AI Timer Control и в списке Properties поменяйте свойство Enabled на True.
7. Сохраните графическую страницу.
8. Верните графическую страницу в режим исполнения. Показания даты и времени будут меняться каждую секунду.