

Class definitions

Contents

- [Overview](#)
- [Creating a class instace](#)
- [Define and call a class method](#)
- [Define classlocal variable](#)
- [Define and use a class property](#)
- [Define and use events within a class](#)
- [Using inherited Methods](#)
- [Using classlocal variables from inherited classes](#)
- [Using inherited Properties](#)
- [Using inherited Events](#)

Overview

Beside predefined components that have been created in C or C++ you may also use object oriented features from within HBasic source files. This document shows some examples how to use this features. To use this features you have to create a new class source file instead of a normal gui based dialog window. Currently you cannot set up GUI components with a class source. This may come with a later version of HBasic. To use class acces select *New* from the menubar or toolbar and select the class type in the dialog that pops up. The first three examples show how to use methods, properties and events from a class source code.

Inheritance within HBasic classes

HBasic already knows a simple form of inheritance for HBasic classes. If you have already defined a class with name class1 and want to inherit from this class you can create a second class definition class2 and start with the source line

Class class2 Inherits class1

You can then use the methods, properties and events defined for class1 with a variable of class2. The example 5 to 8 show examples how you can use methods, properties and examples from an inherited class definition.

Defining and using class methods

A method in a class definition may be declared and called in the same way as in a normal dialog based source (See sub definitions). This example shows how you declare a method within a class and call it from a module instance. When you click on the button of the form the text "Class method called" will be displayed.

```
Class c

  Method m()
    Print "Method called"
  End Method

End Class

' create global instance of class c

Dim gc As c

Sub button1_clicked()
  gc = New c()
  gc.m()
End Sub
```

Example ex_class_method.bas: Example of calling a method within a class definition.

Defining and using classlocal variables

You can create properties for a class source in two different ways. The first is by inserting a simple Dim statement into the body of the class definition and the second defines method within the class source that read or write a value. With methods you can compute the value that should be returned by the property. Normally you need 2 methods for each property, one to read the property (get method) and one to write a value to it (set method). The get-method must return the value of the property and the Let method will be called with the new value as a parameter.

Example how to set up and use properties within a class definition.

```
Class c

  ' l is classlocal property of type long
```

```

    Dim l As Integer

End Class

' create global instance of class c

Dim gc As c

Sub button1_clicked()
    gc = New c()
    gc.l = 1111
    Print gc.l
End Sub

```

Example `ex_classlocal_var.bas`: Use variable defined local in a class.

Defining and using class properties

You can create properties for a class source with subroutine (method) definitions within the class source. Normally you need 2 method for each property, one to read the property (get method) and one to write a value to it (set method). The get-method must return the value of the property and the Let method will be called with the new value as a parameter.

Example how to set up and use properties within a class definition.

```

' Define class with property value

Class c
    Dim i As Integer

    Property val As Integer
        Get
            Return i
        End Get

        Set
            i = value
        End Set
    End Property
End Class

Dim gc As c

' Define instance of new class
' and access class property

Sub button1_clicked()
    gc = New c()

    gc.val = 1234
    Print gc.val
End Sub

```

Example `ex_class_property.bas`: Use property from class definition.

Defining and using class events

To set up an event definition for a class you have to create 3 things

1. Declare an event name with the syntax *Event eventname* in the class definition.
2. Raise the event wherever you like in the class source *RaiseEvent eventname*.
3. Create a subroutine that catches the event with a dialog based source file. The name of the subroutine should be `varname_eventname` where `varname` is the name of the class variable and `eventname` the name of the event that should be caught.

This example shows how you can declare your own events for a class and connect this event definition to a method in the module that declares the class instance.

```

Class c
    Event start_evt()

    Method init()
        RaiseEvent start_evt()
    End Method
End Class

Dim gc As c

Sub button1_clicked()
    gc = New c()
    gc.init()
End Sub

Sub gc_start_evt()

```

```

Print "Event triggered"
End Sub

```

Example ex_class_event.bas: Create class event definition.

Using inherited methods

Compared to the last examples you should change two things to set up code examples with inheritance.

1. Created a definition for a new class *class2* with only one source line "**Inherits** class1".
2. Change all usage of *class1* within the main program to *class2*.

When executing this new programs the result should be the same. If HBasic cannot find the methods, properties and events within the named class *class2* it must search through the inherited class *class1*. This examples only show that this search will find the inherited methods.

```

Class c
    Method m()
        Print "Method called"
    End Method
End Class

Class ci Inherits c
End Class

' Create global instance of class c

Dim gc As ci

Sub button1_clicked()
    gc = New ci()

    ' Call inherited method
    gc.m()
End Sub

```

Example ex_inh_class_method.bas: Call a method of an inherited class

Inherited classlocal variables

This example shows how you can use the variables that have been defined local in a class definition.

```

Class c
    ' l is classlocal property of type long
    Dim l As Integer
End Class

Class ci Inherits c
End Class

' create global instance of class c

Dim gc As ci

Sub button1_clicked()
    gc = New ci()
    gc.l = 1111
    Print gc.l
End Sub

```

Example ex_inh_class_local.bas: Using classlocal variables from an inherited class

Inherited class property

This example shows how you can use the properties from an inherited class definition.

```

' Define class with property value

Class c
    Dim i As Integer

    Property val As Integer
        Get
            Return i
        End Get
    End Property
End Class

```

```

    End Get

    Set
        i = value
    End Set
End Property
End Class

Class ci Inherits c

End Class

Dim gc As ci

' Define instance of new class
' and access class property

Sub button1_clicked()
    gc = New ci()

    gc.val = 1234
    Print gc.val
End Sub

```

Example ex_inh_class_property.bas: Using a class property from an inherited class

Inherited class event

This example shows how you may connect to an event definition of an inherited class event. This needs the same structure as in the normal event definition.

```

Class c

    Event start_evt()

    Method init()
        RaiseEvent start_evt()
    End Method
End Class

Class ci Inherits c

End Class

Dim gc As ci

Sub button1_clicked()
    gc = New ci()
    gc.init()
End Sub

Sub gc_start_evt()
    Print "Event triggered"
End Sub

```

Example ex_inh_class_event.bas: Example of inherited class event