



## Midi Control V1.1

[Copyrights and registering](#)

[Properties](#)

[Events](#)

[Error messages](#)

### Description

The midi control gives you the power to develop MIDI software with Microsoft Visual Basic. It has the possibilities to send and receive ordinary midi messages as well as sending and receiving system exclusive data. Follow the setup guide for installing the software on your system.

### File Name

VBMIDI.VBX

### Object Type

Midi

### Remarks

To develop solid programs, please study this helpfile carefully; e.g. you should examine the ListInDevCount property (only available at runtime) for the amount of midi-in devices, and the ListInDev property which provides you with a list of names of the midi-in devices. Also examine the ListOutDevCount and ListOutDev properties for the midi-out devices. In this way you can make your software work on different machines.



## Copyrights and registration

(C)1995 R.S.Plieger, all rights reserved. The module VBMDEMO.VBX is public domain; it may be copied and distributed for free, as long as it is in its original form (including this helpfile). For evaluation the demoverion works fully for 20 minutes each time you load it.

There are two ways of registering:

- Private registration; send a cheque of US \$35 and your name, address and country to this [address](#) and you will receive the codes for installing your private version of VBMIDI.VBX. This version is NOT to be sold along with software you develop and is only to be used by the person who registered.
- Professional registration; send a cheque of US \$500 to this [address](#), again with name, address and country of your company; you will receive the codes for installing a professional version of VBMIDI.VBX, which also includes a runtime module. This runtime module is only to be sold by the company that registered as part of a midi application developed by that company. Again, the developers module VBMIDI.VBX may NOT be sold and is for developing use only.

The package then can be registered with the [registration program](#).

If you have an e-mail account, please send it along with your name and address, so I can send the registration codes by e-mail and inform you on updates.

**Send to:**

R.S.Plieger  
Postbus 70230  
9704 AE Groningen  
The Netherlands

e-mail: [plieger@bnc.nl](mailto:plieger@bnc.nl)



## Setup

This package includes the following files:

- VBMDEMO.VBX      the demoverion of VBMIDI.VBX
- VBMIDIRT.VBX    the unregistered runtime version of VBMIDI.VBX
- VBMIDI.HLP      the help file
- REGISTER.EXE    the registration program

You should copy the files VBMDEMO.VBX and VBMIDI.HLP to your WINDOWS\SYSTEM directory or to a location your systems PATH is set to in AUTOEXEC.BAT (this way your computer can always find the VBX and the HLP file).

For using it in Visual Basic, start Visual Basic and choose File -> Add file. Now locate the VBMDEMO.VBX and select it; press Ok. The icon of VBMDEMO.VBX should now appear in the Visual Basic toolbar (as a DIN connector). From now on it works like any other Visual Basic object, with its properties and events. The helpfile included is context sensitive, so you can select a property in the property window and press F1 for help on that subject.

The VBMIDIRT.VBX is not registered and therefore does not work. For a working version see the Copyrights and registration section in this helpfile.



## Registration program

When you have registered in either two ways described in the [Copyrights and registration](#) section, you will receive the necessary codes. To register VBMIDI.VBX start the registration program REGISTER.EXE (it has to be in the same directory as VBMDEMO.VBX) and enter the codes exactly as received. Then press <Register> and if all is well the program will confirm that the registration was successful.

If registering a private version, a new file will be created: VBMIDI.VBX. This is your personal version and should only be used by you. You may not sell it or give copies away for free.

If registering a professional version, two files will be created: VBMIDI.VBX in the same directory and VBMIDI.VBX in the directory \RUNTIME. The first is your personal version for developing programs and is not to be sold or given away. The latter is the runtime version and can be distributed as part of the midi software developed by your company; just develop your software with the developers version, and when finished replace it with the runtime version. Pay attention: the runtime version will only work if you use the [RegisterCode](#) property.

## Properties

### *Properties available at designtime and runtime:*

<u>(About)</u>	- Copyrights and registration
<u>Index</u>	- Standard property
<u>Left</u>	- Standard property
<u>MidiInActive</u>	- (De-)activate midi-in device
<u>MidiInDev</u>	- Select midi-in device
<u>MidiOutActive</u>	- (De-)activate midi-out device
<u>MidiOutDev</u>	- Select midi-out device
<u>MidiThru</u>	- (De-)activate midi-thru
<u>Name</u>	- Standard property
<u>Thruchannel</u>	- Set midi-thru channel
<u>TimeOut</u>	- Set time-out for ReceiveExclusive event
<u>Top</u>	- Standard property

### *Properties only available at runtime:*

<u>ListInDev</u>	- List of midi-in devices
<u>ListInDevCount</u>	- Number of midi-in devices
<u>ListOutDev</u>	- List of midi-out devices
<u>ListOutDevCount</u>	- Number of midi-out devices
<u>MidiOut</u>	- Send a single midi-event
<u>Receive</u>	- Start receiving system exclusive data
<u>Request</u>	- Set request data for receiving system exclusive data
<u>SendExclusive</u>	- Send system exclusive data
<u>RegisterCode</u>	- Register module (Runtime version only)

## Events

MidiIn

MTC

ReceiveExclusive

- Receive a single midi-event
- Midi Time Code
- Receive system exclusive data

MIDI  
Musical Instruments Digital Interface

## System Exclusive

Data that cannot be transferred by sending single MIDI events, e.g. parameters of a synthesizer defining its sounds.

## About property

In the demoverison the about property shows how to register to obtain a personal version of VBMIDI.VBX. In a registered version it shows the name and address of the owner. Use by anyone but the official owner is forbidden.

## MidilnActive property

### Description

This property (de-)activates the selected midi-in device set by the MidilnDev property.

### Usage

[*form.*]control.**MidilnActive**[ = *boolean* ]

### Setting

The MidilnActive property settings are:

#### Setting Description

**True** Activates the midi-in device.

**False** (Default)Deactivates the midi-in device.

### Remarks

The internal timer which returns as *TimeStamp* in the Midiln event is set to zero when MidilnActive is set to True.

### Datatype

Integer (Boolean)

## MidiInDev property

### Description

This property selects the midi-in device to receive midi data from.

### Usage

[*form.*]control.**MidiInDev**[ = *integer* ]

### Setting

The MidiInDev property settings are:

Setting	Description
---------	-------------

0 to ( <u>ListInDevCount</u> - 1)	Selects a midi-in device.
-----------------------------------	---------------------------

### Remarks

When another program or another instance of VBMIDI.VBX already opened the selected midi-in device, an ERR\_MIDIDEVINUSE error will occur.

### Datatype

Integer

## MidiOutActive property

### Description

This property (de-)activates the selected midi-out device set by the MidiOutDev property.

### Usage

`[form.]control.MidiOutActive[ = boolean ]`

### Setting

The MidiOutActive property settings are:

#### Setting Description

**True** Activates the midi-out device.

**False** (Default) Deactivates the midi-out device.

### Remarks

When deactivated, an all notes off message is sent to the midi-out device.

### Datatype

Integer (Boolean)

## MidiOutDev property

### Description

This property selects the midi-out device to send midi data to.

### Usage

*[form.]control.MidiOutDev[ = integer ]*

### Setting

The MidiOutDev property settings are:

Setting	Description
0 to ( <u>ListOutDevCount</u> - 1)	Selects a midi-out device.

### Remarks

When another program or another instance of VBMIDI.VBX already opened the selected midi-out device, an ERR\_MIDIDEVINUSE error will occur.

### Datatype

Integer

## MidiThru property

### Description

This property (de-)activates the midi-thru option.

### Usage

```
[form.]control.MidiThru[ = boolean ]
```

### Setting

The MidiThru property settings are:

#### Setting Description

**True** Activates the midi-thru option. If the midi-in device is activated with MidiInActive and the midi-out device with MidiOutActive then all incoming midi-events will be echoed to the midi-out device.

**False** (Default) Deactivates the midi-thru option, no events are echoed.

### Remarks

You can redirect the midi-events to an alternative midi-channel by setting the ThruChannel property.

### Datatype

Integer (Boolean)

## ThruChannel property

### Description

This property selects the midi-thru channel through which the midi-events are echoed.

### Usage

`[form.]control.ThruChannel[ = integer ]`

### Setting

The ThruChannel property settings are:

Setting	Description
0	No channel; all events are send to the original channel
1 to 16	All events are redirected to the selected channel

### Remarks

Note that the midi-in device should be activated with MidiInActive and the midi-out device with MidiOutActive for midi-thru to work.

### Datatype

Integer

# Timeout property

## Description

This property sets the time in periods of 15 ms before a time-out occurs while waiting for a ReceiveExclusive event.

## Usage

`[form.]control.Timeout[ = integer ]`

## Setting

The Timeout property settings are:

Setting	Description
100	(Default) time-out is 1.5 seconds.
20 to 1000	Sets the time-out (0.3 to 15 seconds).

## Remarks

If the software is waiting for system exclusive data which the user has to start manually (e.g. by pressing a dump switch on a synthesizer) it is recommended to use a time-out of say about 5 seconds (Timeout = 333). On the other hand, if a system exclusive dump is requested by your program using the Request function, the time-out should be short, because otherwise the synthesizer can cause a time-out.

## Datatype

Integer

## ListInDev property

### Description

This property is an array of strings containing the names of the midi-in devices. It is not available at design-time and read-only at run time.

### Usage

`[form.]control.ListInDev( Index )`

### Remarks

The index of the first item is 0 and the index for the last item is ListInDevCount - 1. It is very much the same as the standard Visual Basic List property.

### Datatype

String array

## ListInDevCount property

### Description

This property returns the number of midi-in devices. It is not available at design-time and read-only at run time.

### Usage

*[form.]control.ListInDevCount*

### Remarks

It resembles the standard Visual Basic ListCount property.

### Datatype

Integer

## ListOutDev property

### Description

This property is an array of strings containing the names of the midi-out devices. It is not available at design-time and read-only at run time.

### Usage

`[form.]control.ListOutDev( Index )`

### Remarks

The index of the first item is 0 and the index for the last item is ListOutDevCount - 1. It is very much the same as the standard Visual Basic List property.

### Datatype

String array

## ListOutDevCount property

### Description

This property returns the number of midi-out devices. It is not available at design-time and read-only at run time.

### Usage

`[form.]control.ListOutDevCount`

### Remarks

It resembles the standard Visual Basic ListCount property.

### Datatype

Integer

## MidiOut property

### Description

This property sends a single midi-event to the midi-out device.

### Usage

[*form.*]control.**MidiOut**[ = *long* ]

### Setting

The MidiOut property settings are:

Setting	Description
---------	-------------

&Hzzyyx	This value is composed of three elements: &Hxx = midi status byte; &Hyy = first midi data byte; &Hzz = second midi data byte.
---------	---

### Remarks

Note that the midi-out device should be activated with MidiOutActive or MidiOut will return ERR\_DEVNOTACTIVE.

### Datatype

Long

# Receive property

## Description

This property starts the receiving of system exclusive data.

## Usage

[*form.*]control.**Receive**[ = *long* ]

## Setting

The Receive property settings are:

Setting	Description
---------	-------------

1 to ...	This value determines how many system exclusive bytes you want to receive <i>at most</i> ; if you do not know the number of bytes exactly you can specify a very large number. The number of bytes actually received is returned in the <u>ReceiveExclusive</u> event as <i>Length</i> .
----------	--

## Remarks

Note that the midi-in device should be activated with MidiInActive or Receive will return ERR\_DEVNOTACTIVE. Also you should determine the time-out value with the TimeOut property. If you want to receive system exclusive data from a synthesizer by sending a request, first use the Request property, then start the receiving with the Receive property.

## Datatype

Long

# Request property

## Description

This property sets the request data used by Receive. If you want to send a request you cannot use SendExclusive because Receive will stop and start the midi-in device for correct buffer handling. Due to this the internal timer returned as *TimeStamp* in the MidiIn event will be set to zero.

## Usage

[*form.*]control.**Request**[ = *string* ]

## Setting

The Receive property settings are:

Setting	Description
string	You have to compose a string containing the request data.

## Remarks

Note that the midi-out device should be activated with MidiOutActive when using Request, or Receive will return ERR\_DEVNOTACTIVE. If there is no Request, you can use Receive even with the midi-out device being inactive.

## Datatype

String

## SendExclusive property

### Description

This property sends system exclusive data to the midi-out device.

### Usage

[*form.*]control.**SendExclusive**[ = *string* ]

### Setting

The SendExclusive property settings are:

Setting	Description
string	You have to compose a string containing the system exclusive data.

### Remarks

Note that the midi-out device should be activated with MidiOutActive or SendExclusive will return ERR\_DEVNOTACTIVE.

### Datatype

String

## RegisterCode property

### Description

This property is the key for using your runtime module of VBMIDI.VBX (Only available for the professional registered version).

### Usage

`[form.]control.RegisterCode[ = long ]`

### Setting

The RegisterCode property settings are:

Setting	Description
long	Your personal registration code.

### Remarks

This property should be set before using other properties and events of VBMIDI.VBX, or they will not work; e.g. in the Form\_Load() event of Form1. So be carefull to remember your personal registration code!

### Datatype

Long

## Midiln event

### Description

Occurs when a single midi-event was received by the midi-in device.

### Syntax

**Sub** *ctlname\_Midiln* ( [*Index As Integer*,] *Status As Integer*, *Data1 As Integer*, *Data2 As Integer*, *TimeStamp As Long* )

### Remarks

Note that the midi-in device has to be activated by [MidilnActive](#). The Midiln event uses these arguments:

<b>Argument</b>	<b>Description</b>
<i>Index</i>	Uniquely identifies a control if it is in a control array.
<i>Status</i>	Midi status byte; it contains the midi-channel of the midi-event and determines the type of midi-event (e.g. note-on, program-change, etc.)
<i>Data1</i>	First midi data byte; its meaning depends on the status byte.
<i>Data2</i>	Second midi data byte; its meaning depends on the status byte.
<i>TimeStamp</i>	Timing of the midi-event in milliseconds since activating the midi-in device.

## MTC event

### Description

Occurs when the midi-in device is receiving Midi Time Code.

### Syntax

**Sub *ctlname\_MTC* ( [*Index As Integer*,] *MTCType As Integer*, *Hrs As Integer*, *Mins As Integer*, *Secs As Integer*, *Frs As Integer*, *Qfs As Integer* )**

### Remarks

Midi Time Code is for synchronizing your software with other software or devices like videorecorders and multitrack recorders. These machines can have a track with SMPTE time-code on it, and this has to be converted to Midi Time Code. Then you can start the tape at any point, and your software locks on to it. Note that the midi-in device has to be activated by MidiInActive. The MTC event uses these arguments:

<b>Argument</b>	<b>Description</b>
<i>Index</i>	Uniquely identifies a control if it is in a control array.
<i>MTCType</i>	Determines the number of frames per second: 0 = 24 fps 1 = 25 fps 2 = 30 fps drop 3 = 30 fps non drop
<i>Hrs</i>	Hours.
<i>Mins</i>	Minutes.
<i>Secs</i>	Seconds.
<i>Frs</i>	Frames.
<i>Qfs</i>	Quarterframes.

## ReceiveExclusive event

### Description

Occurs when the midi-in device has received system exclusive data or a time-out has occurred while receiving system exclusive data. To start receiving system exclusive data determine the time-out with the TimeOut property and use the function Receive (and if required the function Request).

### Syntax

**Sub** *ctlname*\_ReceiveExclusive( [*Index As Integer*,] *Buffer As String*, *Length As Long* )

### Remarks

The ReceiveExclusive event uses these arguments:

<b>Argument</b>	<b>Description</b>
<i>Index</i>	Uniquely identifies a control if it is in a control array.
<i>Buffer</i>	This is a string containing the received system exclusive data.
<i>Length</i>	This is the length of the string. If <i>Length</i> = 0 then a time-out has occurred.

## Error messages

<u>ERR_DEVOUTOFRANGE</u>	2000
<u>ERR_DEVNOTACTIVE</u>	2001
<u>ERR_MIDI</u>	2002
<u>ERR_MIDIDEVINUSE</u>	2003
<u>ERR_MEMORY</u>	2004
<u>ERR_OUTOFRANGE</u>	2005
<u>ERR_DEMOREADY</u>	2006

ERR\_DEVOUTOFRANGE

2000

Devicenumber out of range. Check number of devices with ListInDevCount and ListOutDevCount.

ERR\_DEVNOTACTIVE 2001

Device not active. Make the device active with MidiInActive or MidiOutActive.

ERR\_MIDI 2002

MIDI error. Something went wrong in the midi-driver.

ERR\_MIDIDEVINUSE            2003

MIDI device is in use. You tried to activate a midi device that was allready activated (by you or by another program).

ERR\_MEMORY 2004  
Out of memory. Not enough memory available for this action.

ERR\_OUTOFRANGE            2005

Parameter out of range. Check out the property description for possible values.

ERR\_DEMOREADY 2006

End of demonstration (demo version only). This message will appear only once after 20 minutes; from then on it is impossible to change the properties.



