



## Help for IniCon

Properties

### Registration Information

### Order Form

### Getting Custom Controls Written

#### **Description**

IniCon is a custom control that makes INI file access simple. No Windows API calls required.

#### **File Name**

INICON3.VBX

#### **Object Type**

IniCon

#### **Compatibility**

VB 2.0 and above

#### **Remarks**

IniCon should be placed on all forms where its functionality is required (including MDI forms and children). IniCon can be placed directly on an MDI form.

**Distribution Note** When you develop and distribute an application that uses IniCon, you should install the file INICON3.VBX into the users Windows SYSTEM directory. IniCon has version information built into it. So, during installation, you should ensure that you are not overwriting a newer version of IniCon.

---

## INI File Layout

If you look in an INI file (WIN.INI is a good example) you'll see many profile strings broken up into application sections. Section headers are formatted as follows:

[Application Name]

In each section there are profile (parameter) strings. Each profile string is identified by a name. These lines are formatted as follows:

Parameter = Value

## Properties

All of the properties that apply to this control are in this table. Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

|                     |                   |               |
|---------------------|-------------------|---------------|
| <u>*Application</u> | <u>Left</u>       | <u>*Value</u> |
| <u>*DeleteApp</u>   | <u>Name</u>       | <u>Width</u>  |
| <u>*DeleteParm</u>  | <u>*Parameter</u> |               |
| <u>Enabled</u>      | <u>Parent</u>     |               |
| <u>*Filename</u>    | <u>Tag</u>        |               |
| <u>Height</u>       | <u>Top</u>        |               |

Value is the default value for the control.

## Application Property

[See Also](#)      [Example \(write\)](#)   [Example \(read\)](#)

### Description

Specifies what part of the INI file to use.

### Usage

[*form.*][*control.*]**Application**[ = *stringexpression* ]

### Remarks

This property determines which application section of an INI file to use. This property must be non-blank.

When a new value is written to this property (and when Parameter is non-blank), a string is read in from the INI file and put into the Value property.

### Data Type

String

## See Also

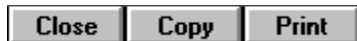
Properties:

[DeleteApp](#)

[Filename](#)

[Parameter](#)

[Value](#)



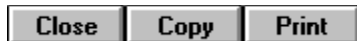
## WIN.INI and Private INI File Writing Example

In this example, the program will write parameters to the IniCon Example section of WIN.INI. The parameter and value properties are determined by two text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, and an IniCon control. Press F5. Then, type a string into each text box and press the button. You can examine the changes to WIN.INI using Notepad or any other text editor.

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    Else
        IniCon1.Filename = ""
        IniCon1.Application = "IniCon Example"
        IniCon1.Parameter = Edit1.Text
        IniCon1.Value = Edit2.Text
    End If
End Sub
```

In this example, the program will write parameters to the IniCon Example section of a private INI file named INIEXAMP.INI. The parameter and value properties are determined by two text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, and an IniCon control. Press F5. Then, type a string into each text box and press the button. You can examine the changes to INIEXAMP.INI using Notepad or any other text editor.

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    Else
        IniCon1.Filename = "INIEXAMP.INI"
        IniCon1.Application = "IniCon Example"
        IniCon1.Parameter = Edit1.Text
        IniCon1.Value = Edit2.Text
    End If
End Sub
```



## WIN.INI and Private INI File Reading Example

In this example, the program will read parameters a user specified section of WIN.INI. The application and parameter properties are determined by two text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, a label, and an IniCon control. Press F5. Then, type the name of the application section you want to examine in the first text box, then type the name of the parameter in the second text box. Finally, press the button and the value will appear in the label.

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    ElseIf Edit2.Text = "" Then
        MsgBox "The second edit box must have text in it."
    Else
        IniCon1.FileName = ""
        IniCon1.Application = Edit1.Text
        IniCon1.Parameter = Edit2.Text
        Label1.Caption = IniCon1.Value
    End If
End Sub
```

In this example, the program will read parameters a user specified section of SYSTEM.INI. The application and parameter properties are determined by two text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, a label, and an IniCon control. Press F5. Then, type the name of the application section you want to examine in the first text box, then type the name of the parameter in the second text box. Finally, press the button and the value will appear in the label.

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    ElseIf Edit2.Text = "" Then
        MsgBox "The second edit box must have text in it."
    Else
        IniCon1.FileName = "SYSTEM.INI"
        IniCon1.Application = Edit1.Text
        IniCon1.Parameter = Edit2.Text
        Label1.Caption = IniCon1.Value
    End If
End Sub
```

## DeleteApp Property

[See Also](#)

[Example](#)

### Description

Deletes an application section from an [INI file](#).

### Usage

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

### Remarks

This property, when set to True, deletes the specified application section from the [INI file](#). The section deleted is determined by the [Filename](#) and [Application](#) properties.

### Data Type

Integer (boolean)

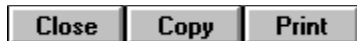


**See Also**

Properties:

[Application](#)

[DeleteParm](#)



## DeleteApp Property Example

In this example, the program will delete a user specified section from WIN.INI. The application property is determined a text box. To try this example, paste the code into the Declarations section of a form that contains a text box, a command button, and an IniCon control. Press F5. Then, type the name of the application section you want to delete in the first text box. Finally, press the button and the application section will be deleted.

**Note:** Be very careful when running this example. This example will delete data from your WIN.INI file. Before running it you should set-up a few sample sections in WIN.INI.

---

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    Else
        IniCon1.FileName = ""
        IniCon1.Application = Edit1.Text
        IniCon1.DeleteApp = True
    End If
End Sub
```

In this example, the program will delete a user specified section from SYSTEM.INI. The application property is determined a text box. To try this example, paste the code into the Declarations section of a form that contains a text box, a command button, and an IniCon control. Press F5. Then, type the name of the application section you want to delete in the first text box. Finally, press the button and the application section will be deleted.

**Note:** Be very careful when running this example. This example will delete data from your SYSTEM.INI file. Before running it you should set-up a few sample sections in SYSTEM.INI.

---

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    Else
        IniCon1.FileName = "SYSTEM.INI"
        IniCon1.Application = Edit1.Text
        IniCon1.DeleteApp = True
    End If
End Sub
```

## DeleteParm Property

[See Also](#)

[Example](#)

### Description

Deletes a parameter entry from an [INI file](#).

### Usage

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

### Remarks

This property, when set to True, deletes the specified parameter section from the [INI file](#). The parameter deleted is determined by the [Filename](#), [Application](#), and [Parameter](#) properties.

### Data Type

Integer (boolean)

**See Also**

Properties:

[DeleteApp](#)

[Parameter](#)

Close

Copy

Print

## DeleteParm Property Example

In this example, the program will delete a user specified parameter from WIN.INI. The application and parameter properties are determined by a pair of text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, and an IniCon control. Press F5. Then, type the name of the application section that contains the parameter you want to delete in the first text box. Then, type the name of the parameter you want to delete in the second text box. Finally, press the button and the parameter will be deleted.

**Note:** Be very careful when running this example. This example will delete data from your WIN.INI file. Before running it you should set-up a few sample sections and parameters in WIN.INI.

---

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    ElseIf Edit2.Text = "" Then
        MsgBox "The second edit box must have text in it."
    Else
        IniCon1.Filename = ""
        IniCon1.Application = Edit1.Text
        IniCon1.Parameter = Edit2.Text
        IniCon1.DeleteParm = True
    End If
End Sub
```

In this example, the program will delete a user specified parameter from SYSTEM.INI. The application and parameter properties are determined by a pair of text boxes. To try this example, paste the code into the Declarations section of a form that contains two text boxes, a command button, and an IniCon control. Press F5. Then, type the name of the application section that contains the parameter you want to delete in the first text box. Then, type the name of the parameter you want to delete in the second text box. Finally, press the button and the parameter will be deleted.

**Note:** Be very careful when running this example. This example will delete data from your SYSTEM.INI file. Before running it you should set-up a few sample sections and parameters in SYSTEM.INI.

---

```
Sub Command1_Click ()
    If Edit1.Text = "" Then
        MsgBox "The first edit box must have text in it."
    ElseIf Edit2.Text = "" Then
        MsgBox "The second edit box must have text in it."
    Else
        IniCon1.Filename = "SYSTEM.INI"
        IniCon1.Application = Edit1.Text
        IniCon1.Parameter = Edit2.Text
        IniCon1.DeleteParm = True
    End If
End Sub
```

## Filename Property

[See Also](#)      [Example \(write\)](#)   [Example \(read\)](#)

### Description

Specifies which INI file to use.

### Usage

[*form.*][*control.*]**Filename**[ = *filename* ]

### Remarks

This property determines if IniCon uses a private INI file or WIN.INI. If this property is blank, profile strings are read from and written to WIN.INI. If this field is not blank, profile strings are written to and read from the file specified by this property.

When a new value is written to this property (and when Application and Parameter are non-blank), a string is read in from the INI file and put into the Value property.

### Data Type

String

**See Also**

Properties:

[Application](#)

[Parameter](#)

[Value](#)

## Parameter Property

[See Also](#)      [Example \(write\)](#)   [Example \(read\)](#)

### Description

Specifies which entry in an application section to use.

### Usage

[*form.*][*control.*]**Parameter**[ = *stringexpression* ]

### Remarks

This property determines which line in an application section of an INI file to use. This property must be non-blank.

When a new value is written to this property (and when Application is non-blank), a string is read in from the INI file and put into the Value property.

### Data Type

String



## See Also

Properties:

[Application](#)

[DeleteParm](#)

[Filename](#)

[Value](#)

## Value Property

[See Also](#)      [Example \(write\)](#)   [Example \(read\)](#)

### Description

Value of the profile string identified by [Filename](#), [Application](#), and [Parameter](#) properties.

### Usage

[*form.*][*control.*]**Value**[ = *stringexpression* ]

### Remarks

This property serves two purposes: to read the value of a profile string from an [INI file](#), and to write a profile string to an [INI file](#). When a program reads this property, the profile string is read from the [INI file](#). When a program writes to this property, the value is written to the [INI file](#).

### Data Type

String

**See Also**

Properties:

[Application](#)

[Filename](#)

[Parameter](#)

## Registration Information

### Credits

IniCon was written by James Shields. Inquiries can be sent to 71231,2066 on CompuServe, or mabry@halcyon.com on Internet. Our mailing address is:

Mabry Software  
Post Office Box 31926  
Seattle, WA 98103-1926

### Registration

You can register this program by sending \$5 (\$10 for international orders) and your address. You can register IniCon **and** its C source code by sending \$25 (\$30 for international orders). With your order, you will receive a copy of our manual documenting all of our controls.

For your convenience, an order form has been provided that you can print out directly from this help file.

### E-mail Discount

You may take a \$2 discount (\$5 for the source code package) for e-mail delivery of this package (CompuServe or Internet). If you choose this option, please note: a printed manual is not included. Be sure to include your full mailing address with your order. Sometimes (on the Internet) the package cannot be e-mailed. So, we are forced to send it through the normal mails.

CompuServe members may also take the e-mail discount by registering this package in the software registration forum (GO SWREG). IniCons SWREG ID number is 833. The source code version's ID number is 1003.

### Credit Card Orders

We accept VISA and Mastercard. If you e-mail your order to us, please be sure to include your card number, expiration date, complete mailing address, and your phone number (in case we have any questions about your order).

© Copyright 1993-1995 by Mabry Software





# IniCon Order Form

Use the Print Topic.. command from the File menu to print this order form.

Mail this form to: Mabry Software  
Post Office Box 31926  
Seattle, WA 98103-1926  
Phone: 206-634-1443  
Fax: 206-632-0272  
BBS: WinDev BBS 206-634-0783  
CompuServe: 71231,2066  
Internet: mabry@halcyon.com

Where did you get this copy of IniCon?

\_\_\_\_\_

Ship to:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Phone:

\_\_\_\_\_

Fax:

\_\_\_\_\_

E-Mail:

\_\_\_\_\_

MC/VISA:

\_\_\_\_\_ exp. \_\_\_\_\_

Disk Size:

(circle one)      3½      5¼

qty ordered \_\_\_\_\_

REGISTRATION  
\$5 each, postpaid (check or money order in hard currency). Outside of North America add \$5.00 shipping.

qty ordered \_\_\_\_\_

SOURCE CODE AND REGISTRATION  
\$25 each, postpaid (check or money order in hard currency). Outside of North America add \$5.00 shipping.

## Getting Custom Controls Written

If you or your organization would like to have custom controls written, you can contact me at the following:

James Shields  
Mabry Software  
Post Office Box 31926  
Seattle, WA 98103-1926  
Phone: 206-634-1443  
Fax: 206-632-0272  
BBS: WinDev BBS 206-634-0783  
CompuServe: 71231,2066  
Internet: mabry@halcyon.com

You can also contact Zane Thomas. He can be reached at:

Zane Thomas  
Post Office Box 300  
Indianola, WA 98342  
CompuServe: 72060,3327

