INI   **Help for IniCon**

Registration Information

Order Form

Getting Custom Controls Written

Description

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

File Name

  INICON3.VBX

Object Type

  IniCon

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 (*).

*Application	Left	*Value
*DeleteApp	Name	Width
*DeleteParm	*Parameter	
Enabled	Parent	
*Filename	Tag	
Height	Top	

Value is the default value for the control.

Application Property

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:

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 <u>parameter</u> and <u>value</u> 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 <u>parameter</u> and <u>value</u> 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

Description

Deletes an application section from an <u>INI file</u>.

Usage

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

Remarks

This property, when set to True, deletes the specied application section from the <u>INI file</u>.   The section deleted is determined by the <u>Filename</u> and <u>Application</u> 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 <u>application</u> 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 <u>application</u> 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

Description

Deletes a parameter entry from an INI file.

Usage

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

Remarks

This property, when set to True, deletes the specied 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

DeleteParm Property Example

In this example, the program will delete a user specified parameter from WIN.INI. The <u>application</u> and <u>parameter</u> 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 <u>application</u> and <u>parameter</u> 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

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

Description

Specifies which entry in an application section to use.

Usage

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

Remarks

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

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

Data Type

String

See Also

Properties:
  <u>Application</u>
  <u>DeleteParm</u>
  <u>Filename</u>
  <u>Value</u>

Value Property

Description

Value of the profile string identified by <u>Filename</u>, <u>Application</u>, and <u>Parameter</u> properties.

Usage

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

Remarks

This property serves two purposes: to read the value of a profile string from an <u>INI file</u>, and to write a profile string to an <u>INI file</u>.   When a program reads this property, the profile string is read from the <u>INI file</u>.   When a program writes to this property, the value is written to the <u>INI file</u>.

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.   If you must send something via U.S. Mail, the address is:

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

Registration

You can register this program by sending $5 ($7 for international orders) and your address. CompuServe members may register by sending $3 and their account number (the registered version will be E-mailed to you).   CompuServe members may also register this package by going to the SWREG forum.   IniCon is registered there.   IniCons registration ID number is 833.

For your convenience, an <u>order form</u> has been provided that you can print out directly from help.

Source Code and Registration

Source code (which includes a registered copy) to this control is available for $25 ($30 for international orders).   With source code you get a registered version of the control.   If you are a CompuServe member, you may get the source code in the Software Registration forum (GO SWREG) for $20.   It's registration number is 1003.

Credit Card Orders

You can order this program with Mastercard, Visa, American Express, or Discover from Public (software) Library by calling 800-242-4775 or 713-524-6394 or by FAX to 713-524-6398 or by CompuServe E-mail to 71355,470.   You can also mail credit card orders to PsL at Post Office Box 35705; Houston, TX 77235-5705.   THESE NUMBERS ARE FOR CREDIT CARDS ONLY.

IniCon's ID number for this service is 11096.   This is good for both the normal registered version, and the source code version.

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, etc., must be directed to Mabry Software at 206-634-1443 or FAX at 206-632-0272.

To ensure that you get the latest version, PsL will notify us the day of your order and we will ship the product directly to you.

© Copyright 1993-1994 by James Shields

INI IniCon Order Form

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

Mail this   Mabry Software
form to: 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: _____

Disk Size: (circle one)   3½   5¼

    qty ordered ____   REGISTRATION
                            $5 each, postpaid (check or money order in hard currency).   Foreign
                            addresses add $2.00 shipping.   No additional shipping charges to Canada or
                            Mexico.

    qty ordered ____   SOURCE CODE AND REGISTRATION
                            $25 each, postpaid (check or money order in hard currency).   Foreign
                            addresses add $5.00 shipping.   No additional shipping charges to Canada or
                            Mexico.

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