

Using the NetVIOS EditBox

Content

The Toolbars
The ToolBar Icons
Toolbar Properties
Calling the Object
Using Style Defaults
Using Language Support
Editing HTML Files In-Place (Content Management Applications)
Other Technical Issues
Appendix

The NetVIOS Editbox is structured as an edit area (iframe) surrounded by toolbars. The toolbars are composed of toolbar icons. Each icon completes an action on the contents of the edit area. The toolbar and toolbar icons are organized so that they may be included or omitted as desired for the function the editbox is intended to perform. The editbox also contains a set of features such as icon color; edit area height and width, as well as the native language (e.g. English or Spanish). All properties and features have intelligent defaults so that it is simply sufficient to declare an editbox with a name. Finally, there are preset combinations of properties that have been determined to save the developer time in

The Toolbars

The NetVIOS editbox natively consists of the following toolbars.

- (1) Format Bar (Default On)
- (2) Insert Bar (Default On)
- (3) Edit Bar (Default On)
- (4) Emoticons (Default Off)
- (5) Toggle Bar (Default On)

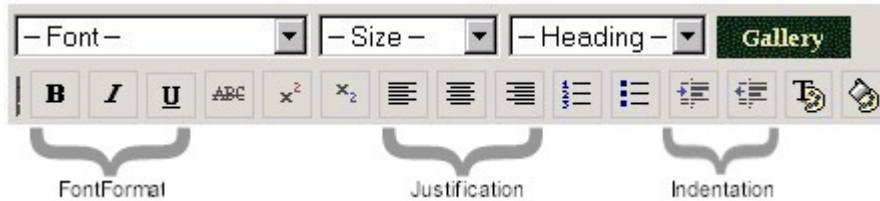
Each of the toolbar may be included or omitted as a property of the editbox. Further, the icons of the toolbars may also be included or omitted as properties. A picture of each toolbar and its contents is presented in the next section:

The item name to use in turning on/off the toolbars (using the **EnableToolBar/DisableToolBar** commands) are "FormatBar", "InsertBar", "EditBar", "Emoticons", and "ToggleBar" respectively. **Item names are not case-sensitive.**

The Toolbar Icons

Format Bar

The icons are arranged such that their properties can be accessed and turned on or off in sensible groups as shown below.



The item names for the toolbar icons (for use with the **DisableToolBarItem** or **EnableToolBarItem** command) are:

FontFormat, Strikethrough, Superscript, Subscript, Justification, Fontcolor, Backcolor, OrderedList, UnorderedList, Indentation, Fontface, FontSize, HeadingFormat, ImageGallery. **Item names are not case-sensitive.**

Insert Bar



The item names of each toolbar item are respectively InsertLink, InsertImage, InsertTable, and InsertRule.

Edit Bar

The toolbar items are grouped as shown below:



The item names of each toolbar item are respectively CutnPaste, UndoRedo, and ContentPrint.

Emoticons



The icons of the emoticon bar are not individually turned on/off. However, the user may disable the entire toolbar by simply clicking its top bar. This collapsible model will be the form of future non-core toolbars added to the NetVIO editbox.

Toggle Bar



The toggle bar contains only one item. Therefore, there is no sub-item to turn on or off. To disallow the user from accessing html, simply disable the entire toolbar [DisableToolBar("ToggleBar")].

ToolBar Properties

Attributes Properties

The attribute properties are the core properties of the editbox and include its name, value, file, language etc.

The **Value** property is the content that is placed within the editbox when drawn. This allows you to use the editbox either for editing or creating new data.

The **File** property is the file that is loaded into the editbox at start allowing you to use the box for content management. The Value property overrides the File property.

The **Language** property accepts ISO 639 International Language Codes. The currently supported languages are English, Spanish, French, German, and Italian. The codes for these languages are "en", "es", "fr", "de", and "it", respectively.

The following are the property handles for the toolbar attributes. Properties that are mandatory are indicated:

- Name (mandatory property)
- NetBoxPath (mandatory for the standalone installation. For the NetVIO installation, this defaults to a folder named "NetBox:" in the NetVIO root).
- Value
- Language
- File
- Language

Appearance

The following are the property handles for the toolbar appearance. All properties have an intelligent default so they do not need to be set.

Width (Default 640 pixels)

Height (Default 200 pixels)

ToolBarColor

ToolBarBoder

ToolBarHover

ToolBarPressed

The last four properties determine what the toolbar icons look like by default. The first variable determines the color, the second the thin border color, while the last two the color when the mouse hovers over, and when pressed, respectively.

Calling the Object

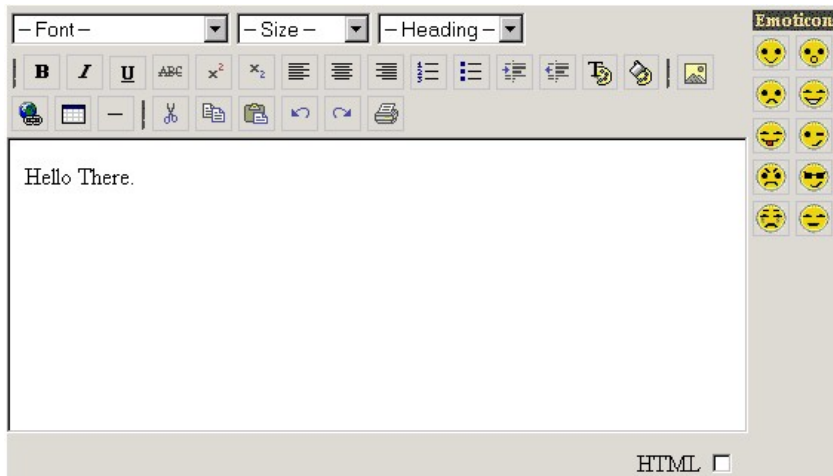
Using NetVIOS within your ASP or ASP.Net script is pretty straightforward:

1. Declare an instance of the EditBox object
2. Set the properties of the editbox to suit your requirements (a list of properties you can set as well as examples are shown in the appendix.
3. Publish or paste the editbox as html on your page using the PublishEditBox method.
4. Create and publish more editboxes (as many as you desire)
5. Undeclare the instance of the Editbox object.

These steps are illustrated in the example below:

1	Dim objedit
2	Set objedit = Server.CreateObject("NetVIOSEditBox.NetBox")
3	objedit.Name = "myfirstbox"
4	objedit.Width = 500
5	objedit.EnableToolBar("Emoticons")
6	objedit.Value = "Hello There."
7	ierror = 0
8	sthestring = objedit.PublishNetBox(ierror)
9	Set objedit = Nothing
10	If (CInt(ierror) = 0) then
11	Response.Write sthestring
12	End if

- (1) Line 1 and 2 declares an instance of the object (**The syntax in Line 2 is for the standalone installation. For the editbox contained as part of NetVIOS, replace "NetVIOSEditBox" with "ubvios"**).
- (2) Lines 3 through 6 sets desired properties
- (3) Line 8 is the command to obtain html (and javascript) strings that will form the editbox. Lines 10 –12 tests that a successful call was made before printing the html string out on the returned html page.
- (4) Line 9 undeclares the instance of the object.



Using Style Defaults

The style defaults are formatted to allow you quickly generate editboxes for difference purposes. The preset styles include – Article, Corporate, Email, Guestbook, Messageboard, Minimal, and Default. To set a style, simply enter the command in the property declaration section:

```
Objedit.NetBoxStyle = "Article"
```

The settings and purpose of each style is explained below:

Article or Content Management Style

This style is intended for use in editing or creating articles. Subsequently, most of the toolbars are turned on except the emoticon toolbar. The width of the editbox is 640 pixels allowing a WYSIWYG environment in which the edited content would look about the same as the final html article or page (for a 640 width website). The following commands will alternatively create this style:

```
DisableToolBar ("Emoticons")  
DisableToolBarItem ("BackColor")  
Width = 640  
Height = 320
```

Corporate Style

Simple editor with conservative colors. Some of the less frequently used icons such as subscripting and superscripting are omitted.

Email

The ToggleBar is turned off here so that the user may not access the raw html syntax. The emoticon toolbar is turned on in this style.

Guestbook

Preferably, the user should also not access raw html with this style. The html toggle bar is turned off here too. In addition, the InsertBar is also turned off.

Minimal

Most of the toolbars are turned off save the font format icons and justification icons. Good for replacing simple textarea tags with a simple html-enrichable editbox.

Using Language Support

The **Language** property of the editbox object accepts ISO 639 International Language Codes. The currently supported languages are English, Spanish, French, German, and Italian. The codes for these languages are "en", "es", "fr", "de", and "it", respectively.

For languages not currently supported, you may change the icon text of any icon to your preferred alternative. Check the appendix for a list of icon text names and the general syntax to replace them. You may send a list of replacements to NetVIOs for a language you would like us to support in future.

Editing HTML Files In-Place (Content Management Applications)

The editbox object will accept a file input and load its html content at the start. This can be done through the **File** property. It is the responsibility of the programmer to strip the returned content of every programming syntax. For ASP sites, this can be done by replacing all "<%" and "%>" segments of the returned string with an empty string.

Technical Issues:

Returning Form Values

When iframes return their contents, the programmer should ensure that the last change or edit initiated by the user is captured. It is a good idea to create a form onSubmit function in javascript (client-side) to capture the last action. Simply insert a call to the javascript editbox function called "*Update_editboxname();*" in your onSubmit function.

Form Validation

Most web programmers prefer to do simple validation on the client side to free up the server resources for more important responsibility. Forms submitted with no value, for instance, should be handled right on the user's own computer. The NetVIOS editbox object can automatically insert a function to test if the editbox has received input. You can then call the function as part of your onSubmit routine. The name of this function is "*Validate_editboxname();*". Check the full syntax for using this feature in the appendix.

Appendix

This appendix contains a full list of addressable properties and methods

PROPERTIES

General Syntax

object.Property = *Value*

The following is a list of all properties. Properties that are bold are mandatory. The File **NetBoxPath** property is mandatory only for the standalone Editbox installation:

The following properties expect string values.

Name

NetBoxPath

Value

File

Language (expects IS 639 language codes. Currently supported codes include en, es, fr, de, and it)

NetBoxStyle (expects values Article, Corporate, Email, Guestbook, Messageboard, Minimal, and Default)

The following properties take on numeric values representing pixel quantities.

Width

Height

The following properties take on color values such as "#000000"

ToolBarColor

ToolBarBorder

ToolBarPressed

ToolBarHover

Icon Text Replacements

The following properties take on string values. The values will replace the default language name of the referenced toolbar icons.

Text_Bold

Text_Italic

Text_Underline

Text_StrikeThrough

Text_Superscript

Text_Subscript

Text_Left

Text_Right

Text_Center

Text_NumberedList

Text_BulletedList

Text_Indent

Text_Outdent

Text_ForeColor

Text_BackColor

Text_Gallery

Text_Image

Text_Link

Text_Rule
Text_Table
Text_Cut
Text_Copy
Text_Paste
Text_Undo
Text_Redo
Text_Print

Text_Happy
Text_Surprised
Text_Sad
Text_Laugh
Text_Chuckle
Text_Wink
Text_Angry
Text_Cool
Text_Cry
Text_Doze

METHODS

The following are methods supported by the NetBox object

PublishNetBox

Syntax

sstring = Object.PublishNetBox (ierror)

Return Values

Html string representing the editbox or an error message

ierror values

- 1 - Name of editbox not set
- 2 - Path of editbox not set

EnableToolBar/DisableToolBar

Syntax

object.EnableToolBar("Itemname")
object.DisableToolBar("Itemname")

Itemname takes on string values (FormatBar, InsertBar, EditBar, Emoticons, ToggleBar)

EnableToolBarItem/DisableToolBarItem

Syntax

object.EnableToolBarItem("Itemname")

object.DisableToolBarItem("Itemname")

Itemname takes on string values ("FormatBar", "InsertBar", "EditBar", "Emoticons", "ToggleBar")

ValidateEditBox

Syntax

Object.ValidateEditBox(fieldlabel, ierror)

Fieldlabel - name to give to error description

Ierror - returns value 1 if the editbox has no name and 2 if it has no path.