



TMS ASP.NET Validation suite Developers Guide

Documentation: Dec, 2009
Copyright © 2009 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email: info@tmssoftware.com

Table of contents

Supported .NET development environments.....	3
Installation	4
Installation of sample applications.....	5
Introduction	6
Overview.....	6
System Requirements.....	6
Class Hierarchy	6
Interfaces.....	7
Assemblies	7
License Agreement	8
TMS ASP.NET Validation controls properties.....	9
TMS TextBox properties list	9
TMS Manager properties list.....	26

Supported .NET development environments

The TMS ASP.NET Validator supports following development environments

Microsoft™ Visual Studio .NET 2005

Microsoft™ Visual Studio .NET 2008

Installation

Installation of components in the IDE

Execute SETUP.EXE

This will install all files under the folder
“\Documents and Settings\All Users\Documents\tmssoftware\”

From the Visual Studio IDE menu, choose

Tools, Customize Toolbox
Switch to the .NET Framework Components tab
Choose Browse and pick TMSValidation.dll from the folder
“\Documents and Settings\All Users\Documents\tmssoftware\”

Installation of sample applications

C# Samples are provided for Visual Studio .NET 2005 and Visual Studio .NET 2008.

Choose: File, Open “Web Site” from the Visual Studio IDE and browse to the folder
\Documents and Settings\All
Users\Documents\tmssoftware\Demos\DemoTMSValidationLibrary” and start the application.

Introduction

Overview

This validation suite made up of two controls: a TMS Textbox input control and a TMS validation Manager. The purpose of this solution is to guide the user to input the correct data. On the server side the control also is able to enforce these rules a second time, in case a browser with JavaScript disabled is used. The validation suite has multiple features or ways to assist the user. These features are called 'Assist Features'. Any of these can be turned on and off. These features make use of an organized hierarchical structure of JavaScript to assist the user on the client side to enter desired values. The amount of JavaScript code will vary vastly depending upon the number of these controls on a page. Each assist feature has three messages associated with it. These messages are used to guide the user. Any number of these messages can be filled in. If any of these messages are left blank, this simply means that method of communication to the user is not to be used. If all messages are set it means that all methods of communication to the user are used.

System Requirements

The following requirements are in effect for this version of Validator Controls

1. A computer running Windows XP, Windows 2003, Windows Vista, Windows 2008 or Windows 7
2. Visual Studio .NET 2005, Visual Studio .NET 2008
3. A version of Internet Information Services (IIS), compatible with the aforementioned Integrated Development Environments (IDE), must be installed and configured for ASP.NET 2.0, ASP.NET 3.5 development
4. A hard drive with at least 15 MB of free space

Class Hierarchy

The TMS Validator Control class hierarchy is as follows.

TMSTextBox
TMSManager
TMSTextBoxDesigner
TMSTextBoxActionList
TMSManagerDesigner
StaticValidator

Interfaces

Defines the following interfaces that may be of use to developers wanting to create custom components and controls for use with this product

IPostBackDataHandler: Defines methods that ASP.NET server controls must implement to automatically load post back data.

IValidator: When implemented by a class, gets or sets the error message text generated when the condition being validated fails.

Assemblies

The TMSValidation.dll assembly is installed into the Global Assembly Cache (GAC):

To complete the installation, the components need to be added to the Visual Studio toolbox.

1. Start Visual Studio .NET 2005 or Visual Studio .NET 2008
2. Create a new WebSite project and switch to the page designer.
3. Create a TMSValidation tab in your toolbox then create one by right clicking and selecting "Add Tab".
4. Right click on the tab name and select "Choose Items..."
5. When the "Choose Toolbox Items" dialog is displayed, Choose Browse and pick TMSValidation.dll from the folder "C:\Documents and Settings\All Users\Documents\tmssoftware\TMS ASP.NET Validation suite"
6. The TMSValidation tab will be filled with the controls

License Agreement

For the registered version, the license agreement for TMSValidation is included as a separate text file in the TMSValidation installation directory. Please see the document at: `\Documents and Settings\All Users\Documents\tmssoftware\TMS ASP.NET Validation suite\single developer license.txt`”.

TMS ASP.NET Validation controls properties

[TMS TextBox properties list](#)

1 Accessibility

TabIndex

This property will set the tabbing order of the control.

Name	Type	ReturnType	Property
Tab Index	Int	Int	Get & Set Property

Remarks: This property will set the tab order of the control

Category: "Accessibility"

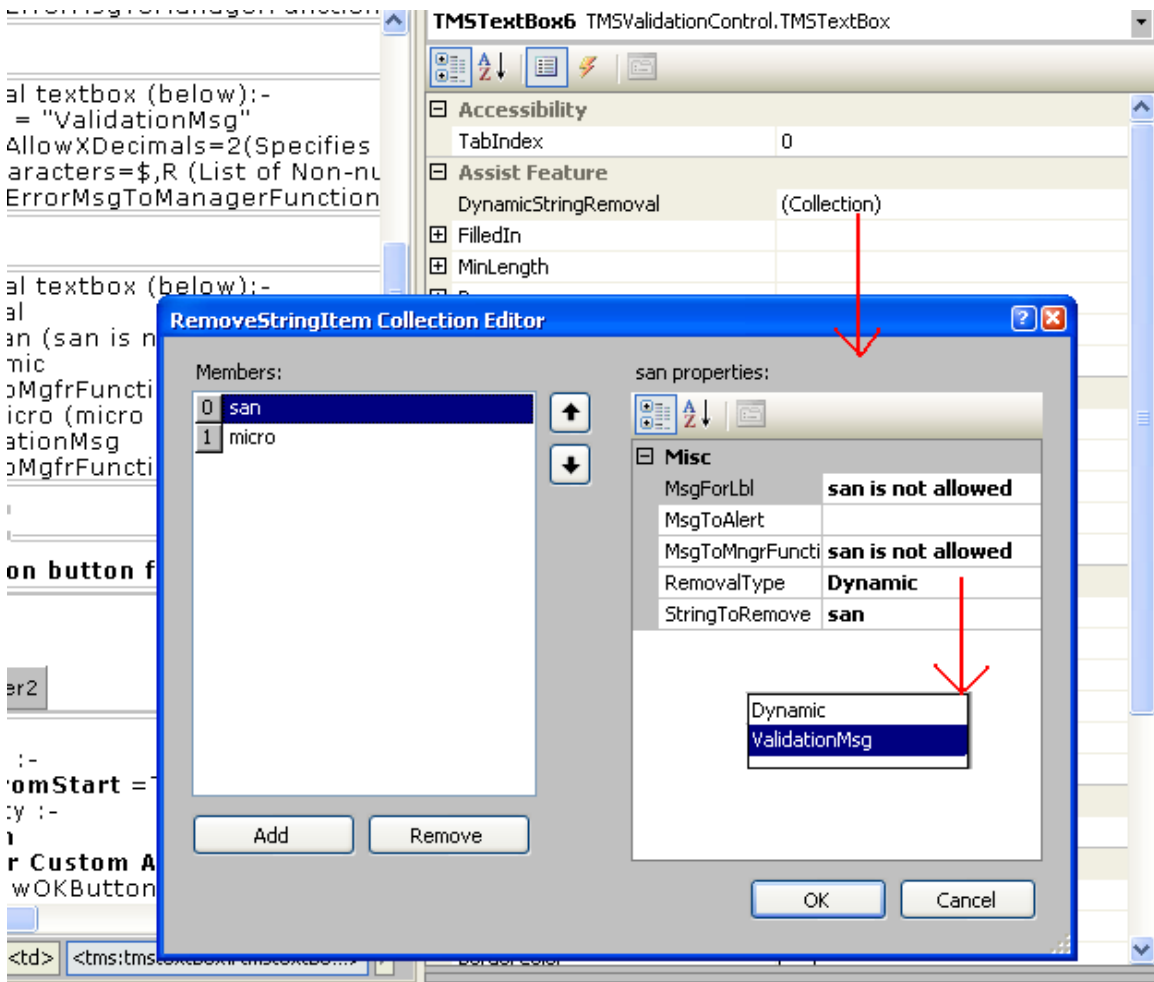
Description: "The tab order of the control"

2 Assist Feature

2.1 DynamicStringRemoval properties

This property provides a collection editor where it will remove the string dynamically or by providing a Validation Message (Removal Type). In Removal type there are two types of options: Dynamic and Validation Msg.

In collection editor, you can add or remove the members (RemoveStringItem) and set the properties as well:



Properties available in RemoveStringItem are:

a) RemovalType:

Name	Type	ReturnType	Property
RemovalType	Enum	Enum	Get & Set Property

i) **Dynamic:** If user selects dynamic, it will remove the string dynamically & will not show any error message.

ii) **ValidationMsg:** If user selects this option, it will show the validation error Message.

b) StringToRemove: User can specify that a particular string will be removed.

Name	Type	ReturnType	Property
StringToRemove	String	String	Get & Set Property

- c) **MsgToAlert:** An “ERROR message” can be specified, to warn the user via a messagebox. It will show the error message in the browser alert box. If this field is empty, no alert message will be shown.

Name	Type	ReturnType	Property
MsgToAlert	String	String	Get & Set Property

- d) **MsgForLbl:** An “ERROR message”, can be specified to warn the user via a label. It will show the error message in the label. If this field is empty, no message will be raised.

Name	Type	ReturnType	Property
MsgForLbl	String	String	Get & Set Property

- e) **MsgToManagerFunction:** An “ERROR message” can be specified that will be returned to the Manager Function.

Name	Type	ReturnType	Property
MsgToManagerFunction	String	String	Get& set Property

Category: "Remove String"

Description: “This Property will remove the string dynamically or provide a Validation Message”.

2.2 FilledIn properties

This holds the settings for a “required field” validation:

FilledIn	
FilledInErrorMsgForLbl	Please enter value in textbox
FilledInErrorMsgToAlert	
FilledInErrorMsgToManagerFun	Please enter value in textbox
ValidateFilledIn	True

2.2.1 FilledInErrorMsgForLbl

This property will flag an error message in a Label if Textbox is not filled in.

Name	Type	ReturnType	Property
FilledInErrorMsgForLbl	String	String	Get & Set

			Property
--	--	--	----------

Remarks: This property holds the text for the validation message via a label.

Category: "Assist Feature"

Description: Error message to show if the user did not complete this field.

2.2.2 FilledInErrorMsgToAlert

This property holds the text for the alert message shown in a message box if textbox is not filled in.

Name	Type	ReturnType	Property
FilledInErrorMsgToAlert	String	String	Get & Set Property

Remarks: This property holds the text for the alert message in the message box, if value is not filled in.

Category: "Assist Feature "

Description: Error message to show if the user does not complete this field.

2.2.3 FilledInErrorMsgToManagerFunction

This property will return back to Manager Function if textbox is not filled in.

Name	Type	ReturnType	Property
FilledInErrorMsgToManagerFunction	string	string	Get & Set Property

Remarks: This property will return to manager function if value is not Filled In.

Category: "Assist Feature"

Description: Message to return to manager function.

2.2.4 ValidateFilledIn

When set to true, the validator will check whether the value is filled in or not. If user selects true, it is a required field. If user selects false in that case, the input for the text box becomes optional, ie. the user can leave the textbox blank.

Name	Type	ReturnType	Property
ValidateFilledIn	bool	Bool	Get & Set Property

Category: "Assist Feature"

Description: Used to validate whether the text is valid, or not

2.3 MinLength properties

Validate that a specific minimum characters have been entered

MinLength	
MinLength	5
MinLengthErrorMsgForLbl	Please enter Minimum 5 Characters
MinLengthErrorMsgToAlert	
MinLengthErrorMsgToManagerFunc	Please enter Minimum 5 Character

2.3.1 Minlength

This property gets or sets the minimum number of characters the user can type or paste in the TMS Textbox Control.

Name	Type	ReturnType	Property
Minlength	int	Int	Get & Set Property

Remarks: The Minimum character of the text that is must.

Category: "Assist Feature"

Description: Minimum required length of a string

2.3.2 MinLenErrMsgToManagerFunction

Name	Type	ReturnType	Property
MinLenErrMsgToManagerFunction	Int	int	Get & Set Property

Remarks: This property will return to manager function if user enters value less than minimum length

Category: "Assist Feature"

Description: Error message returned to Manager Function if validation fails

2.3.3 MinLengthErrorMsgForLbl

Name	Type	ReturnType	Property
MinLengthErrorMsgForLbl	Int	Int	Get & Set Property

Remarks: This property will show an alert message in the label, if user enters value less than minimum length.

Category: "Assist Feature"

Description: Error message to show on label if validation fails

2.3.4 MinlengthErrorMsgToAlert

Name	Type	ReturnType	Property
MinlengthErrorMsgToAlert	Int	Int	Get & Set Property

Remarks: - This property will raise an alert message in a message box, if user enters value less than minimum length

Category “Assist Feature”

Description “Error message to show on alert if validation fails”

2.4 Regex properties

Validation check based on a regular expression

Regex	
RegexErrorMsgForLbl	
RegexErrorMsgToAlert	Please enter Valid Email Address
RegexErrorMsgToManagerFunction	Please enter Valid Email Address
ValidateRegexCode	/^\w+([-\.']\w+)*@\w+([-\.]?\w+)*\.\w+(

2.4.1 RegexErrorMsgForLbl

Name	Type	ReturnType	Property
RegexErrorMsgforLbl	String	String	Get & Set Property

Remarks: This property will flag an error message in label, if regular expression check fails.

Category: “Assist Feature”

Description: Error message to show on label if validation fails

2.4.2 RegexErrorMsgToAlert

This property will raise an alert message in a messagebox if the regular expression check fails.

Name	Type	ReturnType	Property
RegexErrorMsgToAlert	String	String	Get & Set Property

Remarks: This property specifies to raise an alert message in the message box, if user provides invalid data or if validation fails.

Category: “Assist Feature”

Description: Error message to show on alert if validation fails.

2.4.3 RegexErrorMsgToManagerFunction

This property will return back to Manager Function if any error is raised.

Name	Type	ReturnType	Property
RegexErrorMsgToManagerFunction	String	string	Get & Set Property

Remarks: This property will return to Manager Function if any regular expression check fails.

Category: "Assist Feature"

Description: Error message returned to manager function if validation fails.

2.4.4 ValidateRegexCode

Name	Type	ReturnType	Property
ValidateRegexCode	String	String	Get & Set Property

Remarks: This property sets the regular expression.

Example: - /[^]\w+([-+.'\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*\$/g

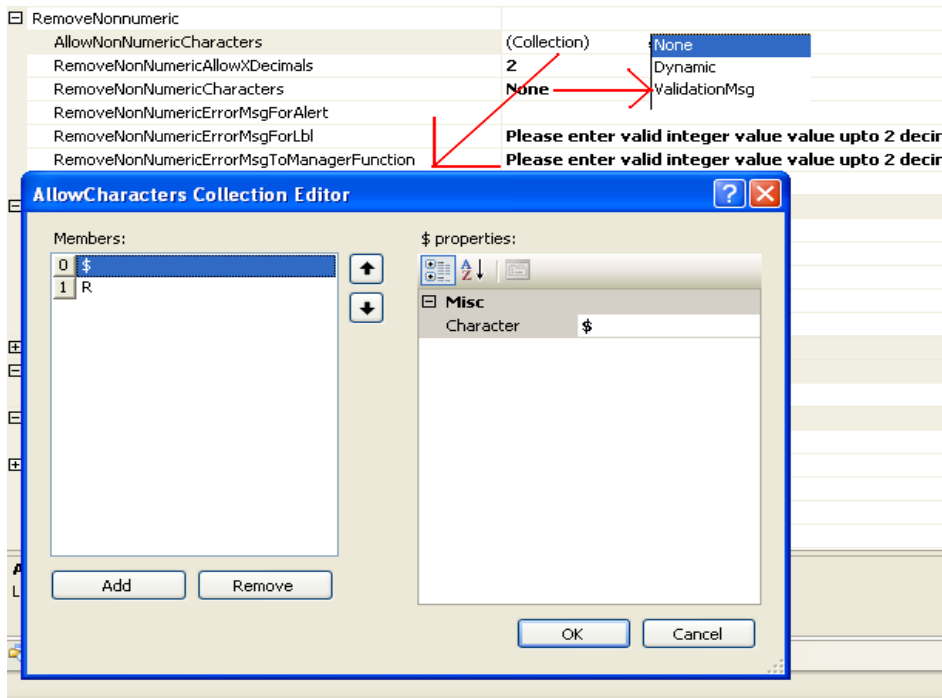
Category: "Validation Features"

Description: Regular expression used to validate text

2.5 RemoveNonNumeric properties

This set of properties allow to perform following validations:

- i. Option to specify number of decimal place to allow. If zero or blank it means no decimal places allowed.
- ii. An array of non-numeric characters to allow. E.g. \$ or R.
- iii. If "ValidationMsg" set in "RemovalType" then it will not dynamically remove the string from textbox. It will show a message in a label or via a messagebox. If "Dynamic" is set in "RemovalType" then it will remove characters dynamically from the textbox and will alert if "MsgForAlert" is set. It will not show the error via a label.



2.5.1 AllowNonNumericCharacters

This property allows to specify an array of non-numeric characters. Example... \$, R, etc.

Name	Type	ReturnType	Property
AllowNonNumericCharacters	Int	int	Get & Set Property

Remarks: This property will allow non numeric characters.

Category: "Assist Feature"

Description: List of non-numeric characters that are allowed

2.5.2 RemoveNonNumericCharacters

i) None: This property will not show any alert message nor remove any characters dynamically.

ii) Dynamic: This will dynamically remove the non-numeric character.(on key press)

iii) ValidationMsg: With this setting, it will alert via a messagebox.

Name	Type	ReturnType	Property
RemoveNonNumeric	Enum	Enum	Get & Set Property

Remarks: This property will remove non-numeric characters.
Category: "Assist Feature"
Description: Option to remove non-numeric characters (Select Type)

2.5.3 RemoveNonNumericAllowXDecimal

This property provides the option to specify number of decimal place to allow. If zero it means no decimal places allowed.

Name	Type	ReturnType	Property
RemoveNonNumericAllowXDecimal	int	int	Get & Set Property

Category: "Assist Feature"
Description: Specifies number of decimal places to allow.

2.5.4 RemoveNonNumericErrMsgForAlert

This property sets the text for alerting via a messagebox.

Name	Type	ReturnType	Property
RemoveNonNumericMsgToAlert	String	string	Get & Set Property

Remarks: This property holds the text to show via a message box. If setting ValidationMsg is set in property RemoveNonNumeric and an incorrect value is entered, it will raise an alert via a messagebox. If the setting Dynamic is set in RemoveNonNumeric, the incorrect value will be dynamically removed without raising an alert.

Category: "Assist Feature"
Description: Error message to show via a messagebox if validation fails

2.5.5 RemoveNonNumericErrMsgForLbl

This property holds the text for a message in a label.

Name	Type	ReturnType	Property
RemoveNonNumericMsgForLbl	String	string	Get & Set Property

Remarks: This property holds the text to show via a label. If setting ValidationMsg is set in property RemoveNonNumeric and an incorrect value is entered, it will show a message in a label. If the setting Dynamic is set in RemoveNonNumeric, the incorrect value will be dynamically removed without showing any error..

Category: "Assist Feature"
Description: Error message to show via a label if validation fails

2.5.6 RemoveNonNumericErrMsgToManagerFunction

This property value will return back to Manager Function if any error is raised.

Name	Type	ReturnType	Property
RemoveNonNumericMsgToAlert	String	string	Get & Set Property

Remarks: This property value will return to ManagerFunction if user provides any wrong information regarding non-numeric entry. If it is desired to show the user a detailed summary list of the errors, this can be configured via the TMSManager (See 4.4 for details).

Category: "Assist Feature"

Description: If user provides wrong information it will return to Manager Function for further assistance.

2.6 ValidateOnChangeFromStart

ValidateOnChangeFromStart False

Name	Type	ReturnType	Property
ValidateOnChangeFromStart	Bool	bool	Get & Set Property

Remarks: When set to true, validation will be done on keypress events, when false validation will be done on focus leave.

Category: "Assist Feature"

Description: Start validation on key press or on focus leave

3) Behavior

3.1 MaxLength

This property will set the maximum required length of string.

Name	Type	ReturnType	Property
MaxLength	int	Int	Get & Set Property

Remarks: This property will set the maximum number of characters in TMS textbox.

Category: "Behavior"

Description: The maximum number of character that can be entered

3.2 ReadOnly

Name	Type	ReturnType	Property
ReadOnly	bool	bool	Get & Set Property

Remarks: This property controls whether the contents of the textbox control can be changed

Category: "Behavior"

Description: Whether the text in the control can be changed or not

3.3 Text Mode

This property will controls the type of entry (single-line, multiline or password) of the textbox control.

Name	Type	ReturnType	Property
Text Mode	String	String	Get & Set Property

Remarks: This property allows to choose 3 text entry modes for the textbox

Category: "Behavior"

Description: Use the text mode property to specify whether a textbox control is displayed as a single-line, multiline, or password textbox. If the textbox control is in password mode, all characters entered in the control are masked.

3.4 ToolTip

Represents a control that creates a pop-up window that displays additional information for the input control.

Name	Type	ReturnType	Property
ToolTip	String	String	Get & Set Property

Category: "Behavior"

Description: The tooltip displayed when the mouse is over the control.

3.5 Visible

Name	Type	ReturnType	Property
Visible	bool	bool	Get & Set Property

Category: "Behavior"

Description: Indicates whether the control is visible and rendered.

4) ErrorLabelAppearance

Holds the settings for the label that will show validation errors.

ErrorLabel Appearance	
ErrorBackColor	<input type="color" value="White"/> White
ErrorBorderColor	<input type="color" value="GrayText"/> GrayText
ErrorBorderStyle	NotSet
ErrorBorderWidth	1
ErrorFont	
ErrorForeColor	<input type="color" value="OrangeRed"/> OrangeRed

Like: -

Please enter Valid numeric Value
 Please enter Valid numeric Value

4.1 ErrorBackColor

Name	Type	ReturnType	Property
ErrorBackColor	Color	Color	Get & Set Property

Category: "Errorlabelappearance"

Description: This property will change the back color of the error label

4.2 ErrorBorderColor

Name	Type	ReturnType	Property
ErrorBorderColor	Color	Color	Get & Set Property

Category: "Errorlabelappearance"

Description: This property will change the border color of the error label

4.3 ErrorBorderStyle

Remarks:- User can set the border style of the label. There are basically ten styles. They are: none, notset, dotted, dashed, solid double groove, ridge, inset, and outset.

Name	Type	ReturnType	Property
ErrorBorderStyle	Enum	Enum	Get & Set Property

Category:- "Errorlabelappearance"

4.4 ErrorFont

Remarks: Allows to set the font of the error message in the label box.

i) Bold

Name	Type	ReturnType	Property
Bold	font	---	Get & Set Property

ii) Italic

Name	Type	ReturnType	Property
Italic	font	---	Get & Set Property

iii) Name

Name	Type	ReturnType	Property
Name	font	---	Get & Set Property

iv) Names

Name	Type	ReturnType	Property
Names	font	---	Get & Set Property

v) Overline

Name	Type	ReturnType	Property
Overline	font	---	Get & Set Property

vi) Size

Name	Type	ReturnType	Property
Size	font	---	Get & Set Property

vii) Strikeout

Name	Type	ReturnType	Property
Strikeout	font	---	Get & Set Property

viii) Underline

Name	Type	ReturnType	Property
Underline	font	---	Get & Set Property

Category: "Errorlabelapperance"

Description: This property is used to change the font of the error label.

4.5 ErrorForeColor

This property is used to set the text color of the error label.

Name	Type	ReturnType	Property
ErrorForeColor	Color	Color	Get & Set

			Property
--	--	--	----------

Category: "Errorlabelapperance"

Description: This property will change the fore color of the error label.

4.6 ErrorBorderWidth

This property is used to set the width of the error label.

Name	Type	Return Type	Property
ErrorBorderWidth	Int	Int	Get & Set Property

Category: "ErrorLabelApperance"

Description: Width of the error label border can be set via this property.

5) Misc

ID

Category: "Misc"

Description: Holds the ID of the control. With this ID, programmatic access to the control is possible.

6) TextBoxAppearance

6.1 BackColor

Sets the background color of the textbox

Name	Type	Return Type	Property
BackColor	Color	Color	Get & Set Property

Category: "Textbox appearance"

Description: "Color of the background of the control"

6.2 BGColor

- 1) Background color when validation fails while user is filling in the TMS Textbox.
- 2) Background color when validation fails when called by the Manager Function.
- 3) Background color when this control has focus.

BgColor	
ManagerValFailBgColor	 0, 192, 0
OnFocusBgColor	 White
ValidationFailBgColor	 128, 64, 64

6.2.1 ManagerValFailBgColor

Name	Type	ReturnType	Property
ManagerValFailBgColor	Color	Color	Get & Set Property

This property will change the background color if manager validation fails

6.2.2 OnFocusBgColor

Name	Type	ReturnType	Property
OnFocusBgColor	Color	Color	Get & Set Property

This property will change the color when the input control gets focus

6.2.3 ValidationFailBgColor

Name	Type	ReturnType	Property
ValidationFailBgColor	Color	Color	Get & Set Property

This property will change the background color if validation fails on the page.

Remarks: This property will change the background color when manager validation fails and change background color on focus change & validation fail.

Category: "Textbox appearance"

6.3 BorderColor

Name	Type	ReturnType	Property
BorderColor	Color	Color	Get & Set Property

Category: "Textbox appearance"

Description: Color of the border around the control

6.4 BorderStyle

Remarks: Sets the border style of the control. There are basically ten styles: none, notset, dotted, dashed, solid double groove, ridge, inset, and outset

Name	Type	ReturnType	Property
BorderStyle	enum	enum	Get & Set Property

Category: "Textbox appearance"

Description: Sets the style of the border around the control

6.5 BorderWidth

This property is used to set the border width of TMS textbox

Name	Type	ReturnType	Property
BorderWidth	int	Int	Get & Set Property

Category: "Textbox appearance"

Description: Width of the border around the control

6.6 CssClass

Name	Type	ReturnType	Property
CssClass	---	---	Get & Set Property

Remarks: CSS Class is used for cascading the style sheets.

Category: " Textbox appearance "

Description: Sets the CSS class name applied to the control

6.7 Font

Remarks: User can set the font of the text in the TMS textbox. The properties are as follows:

i) Bold

Name	Type	ReturnType	Property
Bold	font	---	Get & Set Property

ii) Italic

Name	Type	ReturnType	Property
Italic	font	---	Get & Set Property

iii) Name

Name	Type	ReturnType	Property
Name	font	---	Get & Set Property

iv) Names

Name	Type	ReturnType	Property
Names	font	---	Get & Set Property

v) Overline

Name	Type	ReturnType	Property
Overline	font	---	Get & Set

			Property
--	--	--	----------

vi) Size

Name	Type	ReturnType	Property
Size	font	---	Get & Set Property

vii) Strikeout

Name	Type	ReturnType	Property
Strikeout	font	---	Get & Set Property

viii) Underline

Name	Type	ReturnType	Property
Underline	font	---	Get & Set Property

Category: "Textbox appearance"

Description: The font used for text within the input control

6.8 ForeColor

Name	Type	ReturnType	Property
ForeColor	Color	Color	Get & Set Property

Category: Textbox appearance

Description: Sets the color of the text within the control.

6.9 Height

Name	Type	ReturnType	Property
Height	Int	Int	Get & Set Property

Category: "Textbox appearance"

Description: TSets the height of the control"

6.10 OriginalText

Name	Type	ReturnType	Property
OriginalText	String	String	Get & Set Property

Category: "Textbox appearance"

Description: Holds the original text value of the input control

6.11 Text

This property holds the enforced validated text value of the input control.

Name	Type	ReturnType	Property
Text	string	string	Get & Set Property

Category: "Textbox appearance"

Description: The text value of the input control.

6.12 Width

Name	Type	ReturnType	Property
Width	Int	Int	Get & Set Property

Category: "Textbox appearance"

Description: Sets the width of the control

TMS Manager properties list

1) Alert

1.1 AlertType

- a) Normal
- b) Custom

Alert	AlertType	Custom
Assign TMSTextBoxes	TMSTextBoxesToValidate	Normal
Custom Alert Style		Custom

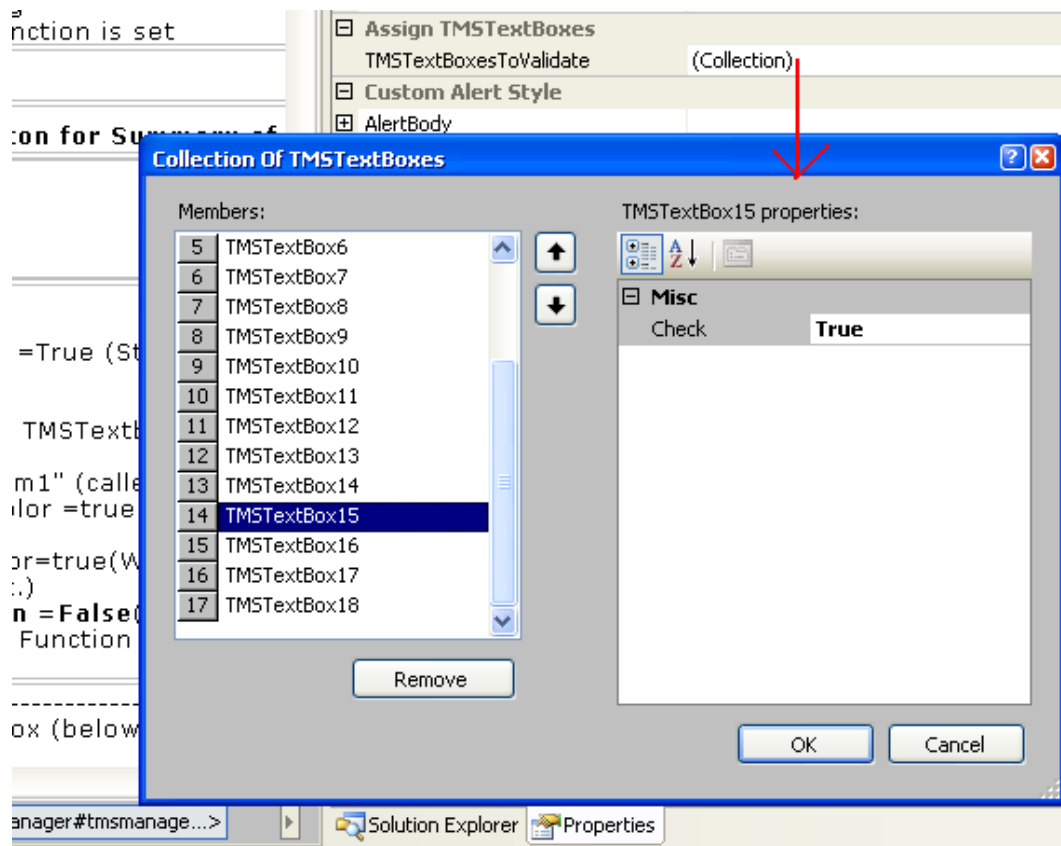
Name	Type	ReturnType	Property
Alert Type	Enum	Enum	Get & Set Property

This property will gives the option to choose whether to use a custom alert or to use the normal alert, ie. the standard browser messagebox.

Category: "Alert"

Description: Selects the alert-type

2) Assign TMSTextBoxes



2.1 TMSTextBoxesToValidate

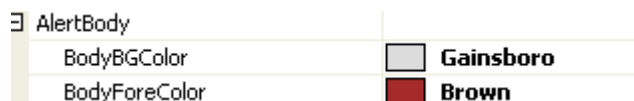
This property shows a collection editor where user can assign TMS text boxes.

Category: "Assign TMStextboxes"

Description: Assign TMStextboxes to the validation manager

3) Custom Alert Style

3.1 - AlertBody



3.1.1 BodyBGColor

This property will set the background color of the custom alert body.

Name	Type	ReturnType	Property
BodyBGColor	Color	Color	Get & Set

			Property
--	--	--	----------

Category: "Custom alert style"
Description: sets the back color of the custom alert



3.1.2 BodyForeColor

This Property will set the forecolor of the custom alert body.

Name	Type	ReturnType	Property
BodyForeColor	Color	Color	Get & Set Property

Category: "Custom alert style"
Description: Sets the fore color

3.2 -AlertTitle

AlertTitle		
IsTitleBar		True
TitleBGColor		IndianRed
TitleForeColor		SeaGreen
TitleText		Validation Error

3.2.1 IsTitleBar

Name	Type	ReturnType	Property
IsTittleBar	bool	Bool	Get & Set Property

Category: "Custom alert style"
Description: This property sets if a titlebar should be visible or not.

3.2.2 TitleBgColor

Name	Type	ReturnType	Property
TitleBgColor	Color	Color	Get & Set Property

Category: "Custom alert style"
Description: This property will set the background color of the titlebar

3.2.3 TitleForeColor

Name	Type	ReturnType	Property
TitleForeColor	Color	Color	Get & Set Property

Category: "Custom alert style"

Description: This property will set the fore color of the titlebar


3.2.4 TitleText

Name	Type	ReturnType	Property
TitleText	String	String	Get & Set Property

Category: "Custom alert style"

Description: Sets the title text of the custom alert

3.3 Border

Border			
AlertBorderStyle		Solid	
BorderColor		 DarkOrange	
BorderWidth		1px	

3.3.1 AlertBorderStyle

Name	Type	ReturnType	Property
BorderStyle	enum	Enum	Get & Set Property

Category: "Custom alert style"

Description: Selects the border style of custom alert

3.3.2 BorderColor

Name	Type	ReturnType	Property
BorderColor	Color	Color	Get & Set Property

Category: "Custom alert style"

Description: Sets the border color"

3.3.3 Borderwidth

Name	Type	ReturnType	Property
Borderwidth	Int	Int	Get & Set Property

Category: "Custom alert style"

Description: Sets the border width

ShowCloseButton	True
ShowOKButton	True

3.3 ShowCloseButton

Name	Parameter	Type	ReturnType	Property
ShowCloseButton		bool	bool	Get & Set Property

Category: "Custom alert style"

Description: Controls whether to show exit button on the top right of custom alert or not. If this property is set to **false** then exit button will not be shown

3.4 ShowOKButton

Name	Parameter	Type	ReturnType	Property
ShowOKButton		String	String	Get & Set Property

Category: "Custom alert style"

Description: Controls whether the user can exit the custom alert style by clicking an OK button

3.4 AlertFont

AlertFont	14px
Bold	False
Italic	False
Name	<input type="text"/>
Names	
Overline	False
Size	14px
Strikeout	False
Underline	False

Remarks: Allows to set the font of the text in the custom alert:

i) Bold

Name	Type	ReturnType	Property
Bold	font	---	Get & Set Property

ii) Italic

Name	Type	ReturnType	Property
Italic	font	---	Get & Set Property

iii) Name

Name	Type	ReturnType	Property
Name	font	---	Get & Set Property

iv) Names

Name	Type	ReturnType	Property
Names	font	---	Get & Set Property

v) Overline

Name	Type	ReturnType	Property
Overline	font	---	Get & Set Property

vi) Size

Name	Type	ReturnType	Property
Size	font	---	Get & Set Property

vii) Strikeout

Name	Type	ReturnType	Property
Strikeout	font	---	Get & Set Property

viii) Underline

Name	Type	ReturnType	Property
Underline	font	---	Get & Set Property

Category: "Custom alert style"

Description: Sets the font used for text within the alert

4) Management Functionality

Management Functionality	
ErrorRightsOfMainFunction	False
IsManagerfailChangeBgColor	True
IsValidationfailChangeBgColor	True
ShowSummary	True

4.1 ErrorRightsOfMainFunction

Name	Type	ReturnType	Property
ErrorRightsOfMainFunction	Bool	bool	Get & Set Property

Remark: This property sets the error rights of the main function. If this property is set to true it means the validation error message will be shown by the main function or not.

Category: "Management Functionality"

Description: Defines whether to show errors by the central manager function or not

4.2 IsManagerfailChangeBgColor

Name	Type	ReturnType	Property
IsManagerfailChangeBgColor	Bool	bool	Get & Set Property

Remark: This boolean property sets whether to change background color of the TMS textbox on the failure of the validation in the scope of the Manager Function. If this property is set to true it means background color will be changed else not.

Category: "Management Functionality"

Description: Sets whether to change background color of the TMStextbox on manager validation fail or not

4.3 IsValidationfailChangeBgColor

Name	Type	ReturnType	Property
IsValidationfailChangeBgColor	Bool	bool	Get & Set Property

Remark: This boolean property sets whether to change background color of the TMS textbox on the failure of the validation in the scope of the main function. If this property is set to true it means background color will be changed else not.

Category: "Management Functionality"

Description: Sets whether to change background color of the TMStextbox on validation fail or not

4.4 ShowSummary

Name	Type	ReturnType	Property
ShowSummary	bool	Bool	Get & Set Property

Remark: This boolean property sets whether to show summary of all the error related to the manager or not. If this property is set to true it means summary will be shown else not.

Category: "Management Functionality"

Description: Sets whether to show errors summary or not

6) TMS Manager

TMS Manager	
(ID)	TMSManager2
ManagerFunctionName	San1

6.1 (ID)

Category: "TMS Manager"

Description: Sets the name of the control

6.2 ManagerFunctionName

Name	Type	ReturnType	Property
ManagerFunctionName	string	string	Get & Set Property

Category: "TMS Manager"

Description: "Name for the Manager Function. Note the Manager Function is available but it is up to the developer to call the Manager Function on click of button or else For Summary"

```
<asp:Button ID="Button1" runat="server" Text="Custom Alert" OnClientClick="return San1()" />
</asp:Button>
```