

CTC ASP.NET WebForms Configurator

Version 2.0.3



Table of Contents

1	Introduction	3
1.1	What is CTC ASP.NET WebForms Configurator?	3
1.2	The Concept	3
1.3	Configurable Elements	3
1.4	License Keys	4
1.5	Standard Controls.....	4
2	Configuration Items	5
2.1	Custom Controls.....	5
2.2	Control Substitutions, Add.....	6
2.3	Control Substitutions, Remove.....	8
2.4	Control Specifications, Add.....	9
2.4.1	Control Options	11
2.5	Control Specifications, Remove.....	13
2.6	Font Substitutions, Add.....	14
2.7	Font Substitutions, Remove.....	16
2.8	Options	17
2.9	Runtime Configuration	23

1 Introduction

1.1 What is CTC ASP.NET WebForms Configurator?

The CTC ASP.NET WebForms Configurator is the Configurator for the ASP.NET Web Forms Generator from Client Tools Consultancy.

The Configurator is an add-in to the CTC Configurator Framework. It provides the user interface to configure options and features specifically for the ASP.NET WebForms Generator.

This document should be read in conjunction with the **CTC Configurator Framework** document and the **CTC ASP.NET WebForms Generator** document.

1.2 The Concept

Typically, a generator creates a fixed, predetermined user interface application, where users have no or limited influence on what is generated, and customizations have to be applied by modifying the generated user interface application, or by writing a custom generator.

The concept of CTC Generators is to include as many requirements as possible in the generate stage rather than applying modifications to the User Interface after it has been generated.

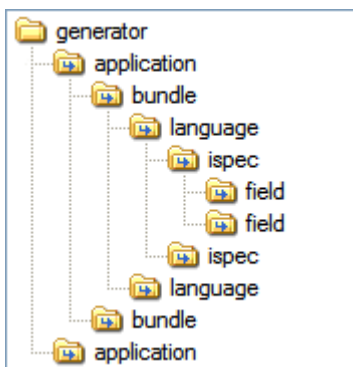
This requires the generator to be very flexible and to provide the means for customizing the result of the generate prior to entering the generate phase.

CTC Generators provide the ability to influence what is generated by configuring features, setting options, customizing Standard Controls, adding Custom Controls and substituting Controls. The generated user interface is still based on the forms and controls being painted in the EAE or AB Suite development environment, however, the CTC ASP.NET WebForms Configurator allows the developers to specify how each form and control is to be generated at the application, bundle, language, ispec and field level.

Being able to configure and specify the required customization before the generate phase provides a repeatable and automated solution that in most cases will not need further modifications to the generated source.

1.3 Configurable Elements

A configuration consists of configurable elements, and options and features that can be specified for each of these elements. Configurable elements are generator, application, bundle, language, ispec and field, which are specified in a hierarchy as illustrated below.



For information about configurable elements and their hierarchical relationship, refer to the help documentation for the **CTC Configurator Framework**.

The following is a list of items (options/features) that can be configured for the ASP.NET WebForms Generator within each of the configurable elements (generator, application, bundle, language, ispec and field):

- License Keys
- Custom Controls
- Control Substitutions
- Control Specifications
- Options
- Runtime Configuration

1.4 License Keys

License Keys are applicable to the generator and can be specified at the generator level only.

License keys for the ASP.NET WebForms Generator are specific for the computer on which the generator is installed. A license key is obtained by contacting Client Tools Consultancy.

License keys are stored in this element. When starting the ASP.NET Generator it will request the user to enter a valid license key, evaluation license or full license. Alternatively, the license key can be entered directly into the configuration by adding a License key element to the ASP.NET Generator.

When sharing configuration files by copying a configuration file, the license key element can contain keys that are invalid on the current computer. Invalid license keys are ignored by the generator.

Invalid or expired license keys can be deleted.

1.5 Standard Controls

Standard Controls provide the default look and feel of the controls as they are painted and specified in the EAE and AB Suite Development environments. Standard Controls are built into the ASP.NET WebForms Generator. They can be used as they are without further customization.

The following is the list of Standard Controls available with the ASP.NET WebForms Generator:

Control	Description
ButtonGroup	Container for single button controls such as Check Box, Push Button and Radio Button. AB Suite only.
CheckBox	Single button Check Box. AB Suite only.
CheckBoxList	List of Check Boxes, horizontal/vertical. EAE 3.3 only.
ComboBox	Drop down list box.
Form	Container for controls painted on a form.
Form Page	Outermost container for the form.

Image	Image.
Label	Label.
Line	Box and horizontal and vertical line.
ListBox	List Box.
ListDataSource	DataSource control for ListBox and ComboBox controls.
ListXmlDataSourceControl	DataSource control for static ListBox and ComboBox controls.
Panel	Group container for other controls. AB Suite only.
PushButton	Single Push Button. AB Suite only.
PushButtonList	List of Push Buttons, horizontal/vertical. EAE 3.3 only.
RadioButton	Single Radio Button. AB Suite only.
RadioButtonList	List of Radio Buttons, horizontal/vertical. EAE 3.3 only.
Table	Table container for CopyFrom regions.
TableHeaderRow	Table Header Row container for CopyFrom regions.
TableHeaderCell	Table Header Cell container for CopyFrom regions.
TableRow	Table Row container for CopyFrom regions.
TableCell	Table Cell container for CopyFrom regions.
Teach	Container for controls painted on a teach form.
Teach Page	Outer container for the teach form.
TeachText	Text for the teach form.
TextBox	Text input box.

2 Configuration Items

2.1 Custom Controls

Custom Controls are controls created by either the customer or CTC for a specific purpose. A Custom Control is used when none of the Standard Controls match the requirements.

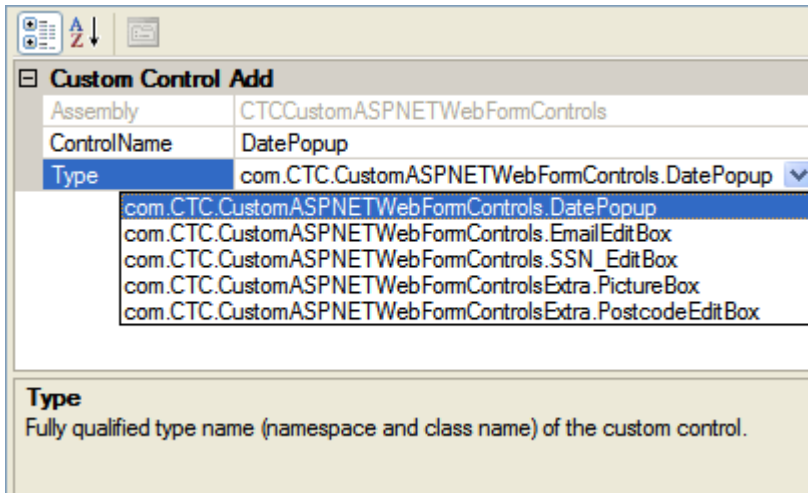
Custom Controls also allow the utilization of third party controls in the generated user interface application. For example, to utilize a Date Picker control from a third party, a custom control can be created as an add-in to the generator, which would generate the correct source specifications and code for the third party control to be executed at runtime in the generated user interface application.

For information on how to create a Custom Control, refer to the **CTC ASP.NET WebForms Generator** documentation.

A Custom Control is added to the configuration at the generator level only and applies to all configurable items such as application, bundle, ispec and field items within the generator.

The assembly dll containing the Custom Control must be copied to the bin sub-folder of the Component Enabler Root directory, for example, C:\CE3.3\bin.

When adding a Custom Control, the ControlName and Type properties as illustrated below must be specified.



Assembly	CTCCustomASPNETWebFormControls
ControlName	DatePopup
Type	com.CTC.CustomASPNETWebFormControls.DatePopup

Type
Fully qualified type name (namespace and class name) of the custom control.

Type: This is selected from the dropdown list, which is automatically filled with a list of Custom Controls found within the bin sub-folder of the Component Enabler Root directory. The type specifies the fully qualified type name, which is the namespace and class name of the Custom Control.

ControlName: User specified unique name of the Custom Control. When adding a new control type, this is pre-filled with the type name of the control.

Assembly: This identifies the assembly dll which contains the Custom Control. This is automatically filled when selecting the type.

ControlDllLocation: This specifies the path to where the dll of the third party control is installed. This property is only required when the Custom Control is implementing a control from a third party such as ComponentArt.

2.2 Control Substitutions, Add

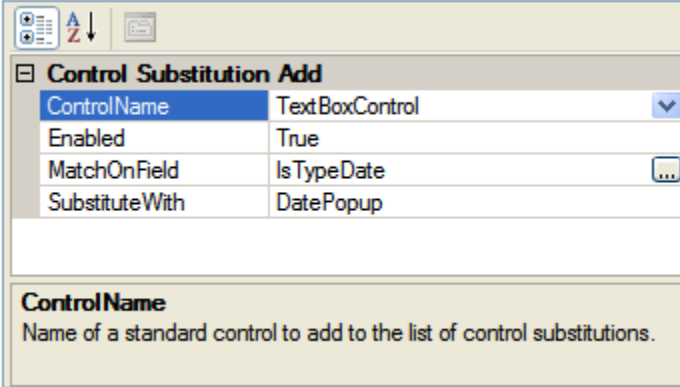
Control Substitutions are used when replacing/substituting Standard Controls with Custom Controls.

Control Substitutions can be added to any level in the hierarchy.

Controls being substituted must be of similar type. For example, when substituting a textbox control with a custom date popup control, the date popup control must be capable of understanding how to generate the control that was originally painted as a textbox.

When control substitutions are inherited from parent levels, control substitutions at the current level are evaluated first. When substituting the same control with different custom controls specifying match conditions, the substitutions will be evaluated in the order they are specified, except for unconditional substitutions which will be evaluated last.

When adding a Control Substitution, the ControlName and SubstituteWith properties as illustrated below must be specified.



Control Substitution Add	
ControlName	TextBoxControl
Enabled	True
MatchOnField	Is TypeDate
SubstituteWith	DatePopup


ControlName
Name of a standard control to add to the list of control substitutions.

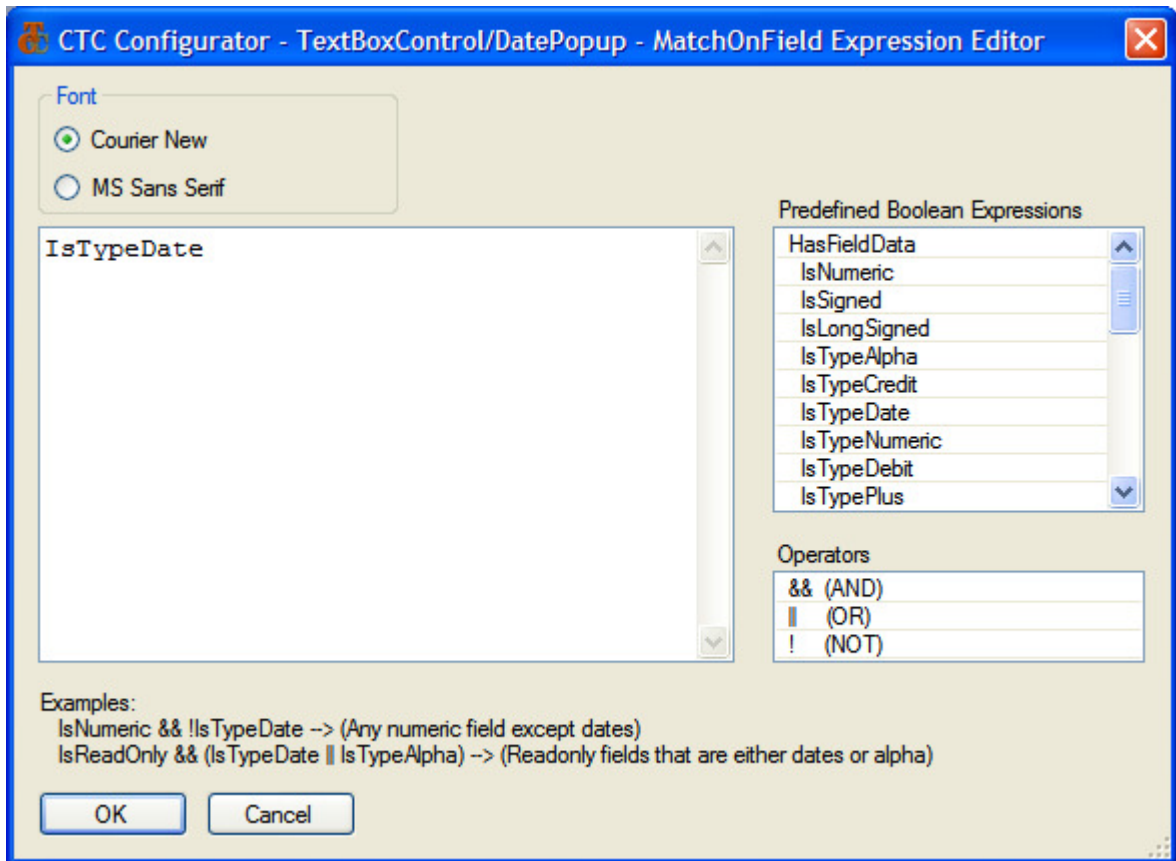
ControlName: This specifies the name of a Standard Control to be substituted. This is selected from the dropdown list, which is automatically filled with a list of available Standard Controls.

Enabled: This property provides a convenient way of removing a control substitution from the generate process without deleting it.

SubstituteWith: This specifies the name of a Custom Control to use instead of the Standard Control. This is selected from the dropdown list, which is automatically filled with a list of Custom Controls currently added to the generator.

MatchOnField: Optional match expression. MatchOnField is a boolean expression specifying the condition for selecting to which field or fields to apply the substitution. At generate time, the MatchOnField expression is evaluated against the field associated with the particular control and when the expression evaluates true, the substitution is applied. When no expression is specified, control substitution is unconditionally applied.

The expression is specified by selecting the ellipsis button . This will open the Expression Editor window as illustrated below.



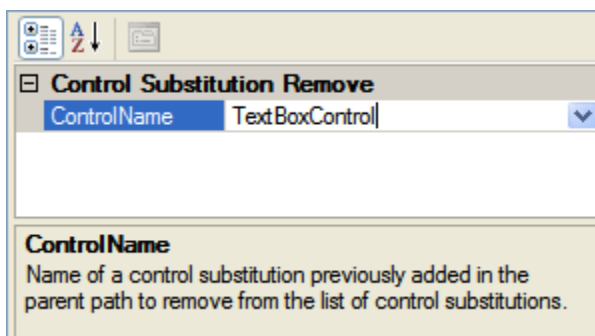
An expression is specified using a selection of predefined boolean expressions. Expressions are dragged/dropped or copied/pasted from the list of Predefined Boolean Expressions. Operators such as && (AND), || (OR), ! (NOT), == (Equal), != (Not Equal), > (Greater Than), < (Less Than), >= (Greater Than or Equal) or <= (Less Than or Equal) as well as parentheses () can be used to create complex expressions.

2.3 Control Substitutions, Remove

A Control Substitution that has previously been added to a parent in the hierarchy can be removed at any level in the hierarchy by inserting a Control Substitution Remove item.

For example, if a substitution of TextBoxControl with DatePopup has been added at the application level, it may be desirable to generate a specific field as a standard TextBoxControl and not as a DatePopup. In this case, a Control Substitution Remove item can be added to the field specifying to remove the substitution for the TextBoxControl.

The properties window of Control Substitution Remove is illustrated below.



ControlName: This specifies the name of a Standard Control that previously has been substituted at a parent level in the hierarchy. This is selected from the dropdown list, which is automatically filled with a list of available Standard Controls. For convenience, a 'Remove All' item is available in the list. When selecting 'Remove All', all control substitutions inherited from the parent levels will be removed.

2.4 Control Specifications, Add

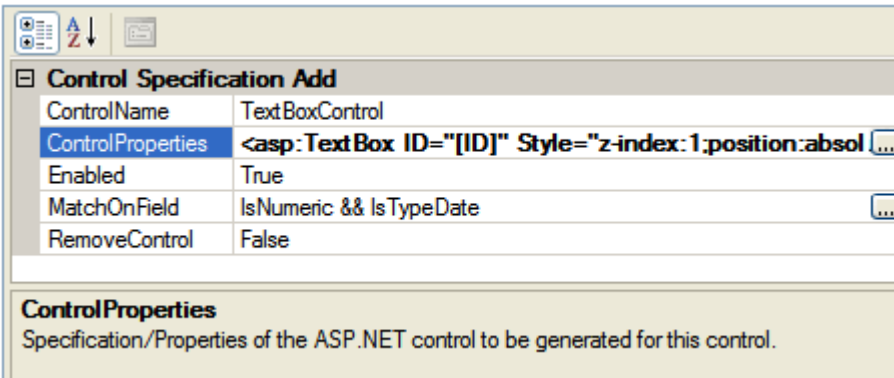
Control Specifications are used to change the default specifications of Standard Controls and Custom Controls. This allows modification of the default look and feel of a control by adding or removing properties and style attributes.

The specification of a control specifies the look and feel of the control. The properties of the control as it is generated are specified here.

Control Specifications can be added to any level in the hierarchy.

When control specifications are inherited from parent levels, control specifications at the current level are evaluated first. When specifying the same control with different match conditions, the specifications will be evaluated in the order they are specified, except for unconditional specifications which will be evaluated last.

When adding a Control Specification, the ControlName and ControlProperties as illustrated below must be specified.

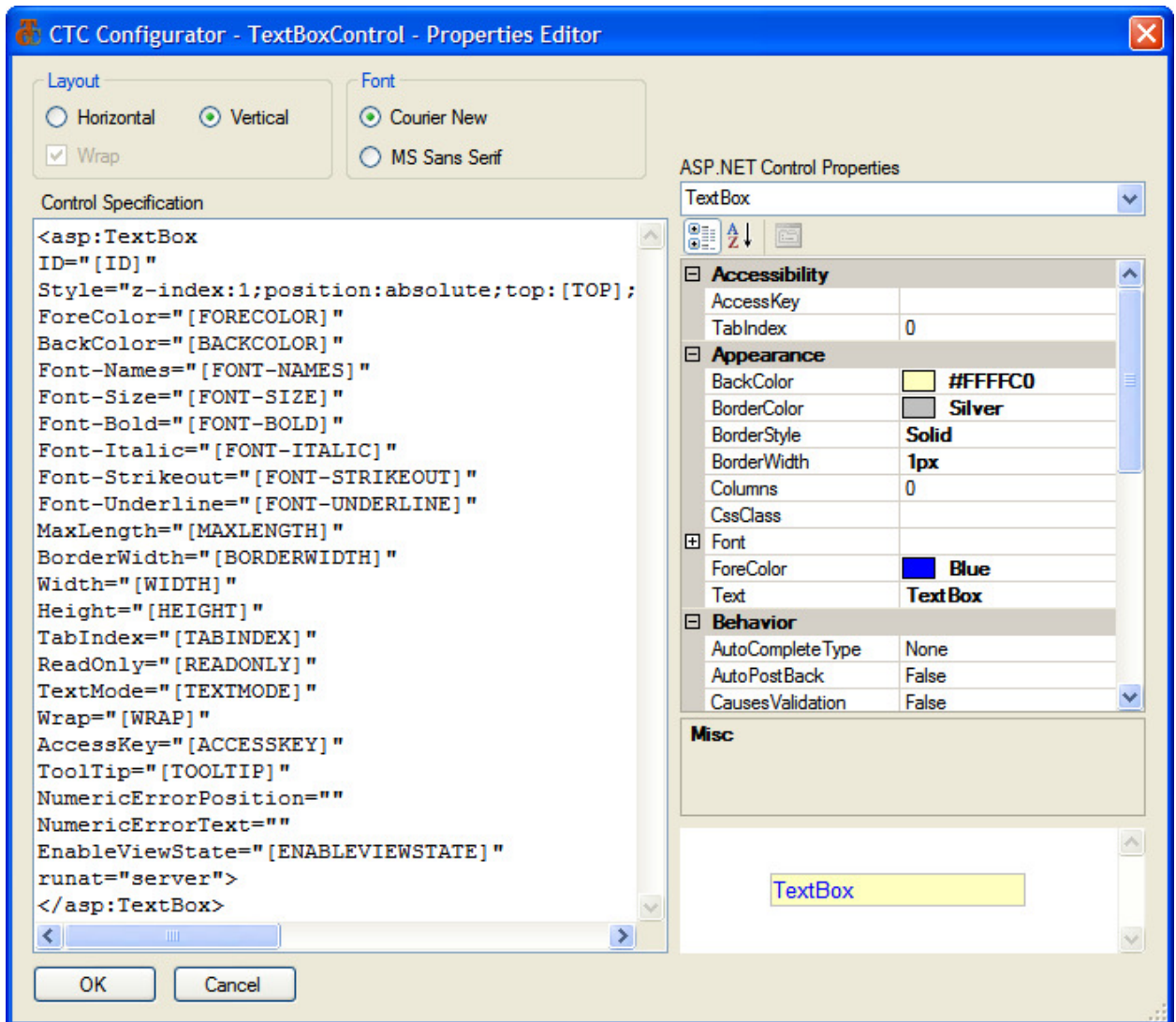


Control Specification Add	
ControlName	TextBoxControl
ControlProperties	<asp:TextBox ID="[ID]" Style="z-index:1;position:absol ...
Enabled	True
MatchOnField	IsNumeric && IsTypeDate
RemoveControl	False

ControlProperties
Specification/Properties of the ASP.NET control to be generated for this control.

ControlName: This specifies the name of a Standard Control or a Custom Control for which to specify properties. This is selected from the dropdown list, which is automatically filled with a list of available Standard Controls and Custom Controls.

ControlProperties: This specifies the properties of the control. When selecting a control from the dropdown list of the ControlName property, the default specifications of the control will automatically be shown here. The properties are edited by selecting the ellipsis button (...) which will open the Control Properties Editor window as illustrated below.



The specification of a control specifies exactly how the control is generated. Via this editor, the user specifies the properties and their value to be set for the control.

The properties that can be specified on a control are those defined for the Microsoft ASP.NET controls. Any property available on an ASP.NET control can be added to the specifications of a CTC Standard Control.

The variables specified in square brackets represent the equivalent properties as they are specified for a control when painted in EAE or AB Suite. At generate time, the generator will replace the variables in square brackets with their actual value. If a variable has no value, i.e. it has not been specified in EAE or AB Suite, the whole property will be removed from the control at generate time.

The variables available for a control depend on the control. When initially selecting a control from the ControlName property, the default specifications of the control show the complete set of variables and their usage for the control as seen in the editor window above.

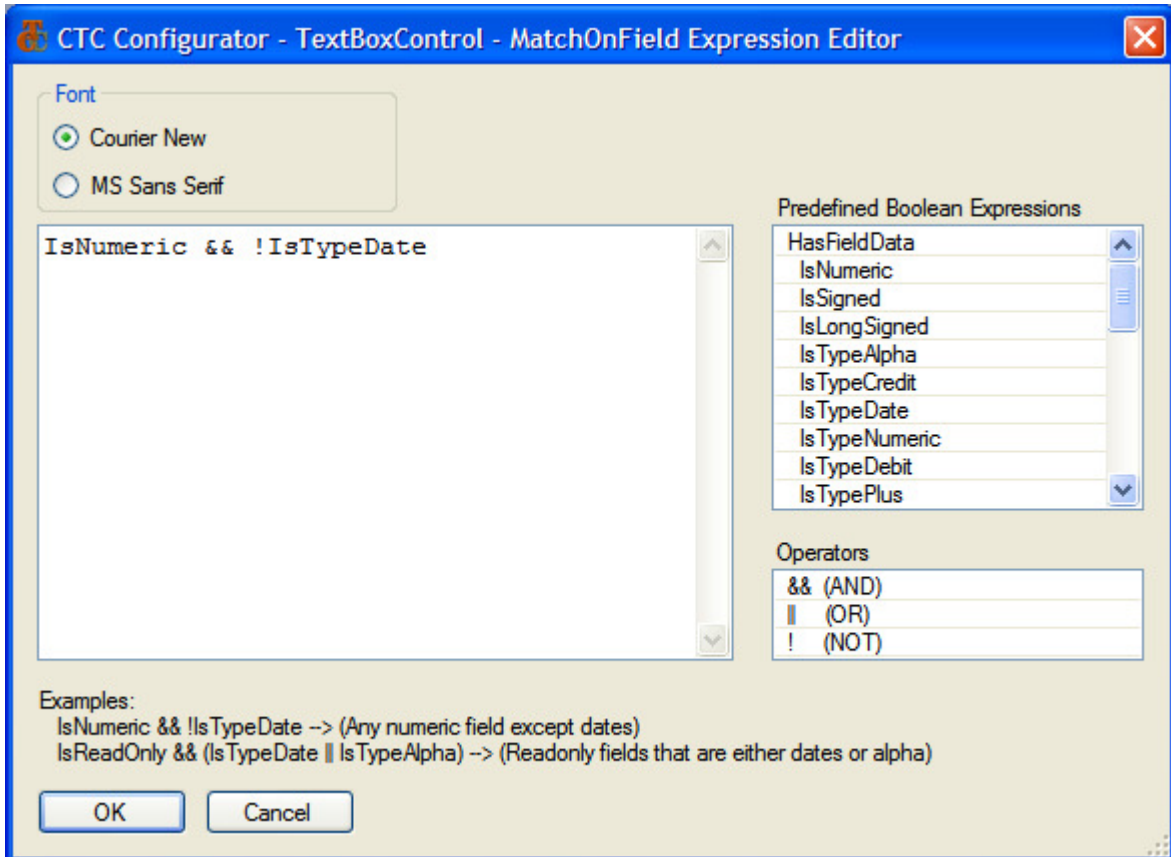
The variable names are case sensitive.

For easy readability, the editor window includes options for the user to choose layout and font.

Enabled: This property enables/disables the configuration of the specifications for a control. When false, the generator ignores this configuration entry. It provides a convenient way of removing a control specification from the generate process without deleting it.

MatchOnField: Optional match expression. MatchOnField is a boolean expression specifying the condition for selecting to which field or fields to apply the substitution. At generate time, the MatchOnField expression is evaluated against the field associated with the particular control and when the expression evaluates true, the substitution is applied. When no expression is specified, control substitution is unconditionally applied.

The expression is specified by selecting the ellipsis button  which will open the Expression Editor window as illustrated below.



An expression is specified using a selection of predefined boolean expressions. Expressions are dragged/dropped or copied/pasted from the list of Predefined Boolean Expressions. Operators such as && (AND), || (OR), ! (NOT), == (Equal), != (Not Equal), > (Greater Than), < (Less Than), >= (Greater Than or Equal) or <= (Less Than or Equal) as well as parentheses () can be used to create complex expressions.

RemoveControl: Specifies whether this control is to be removed from the form. This is a convenient way of excluding a control from being generated.

2.4.1 Control Options

In addition to the ASP.NET control properties, the following CTC Standard Controls have been extended with properties as options for the generation of behaviours specific to the CTC Standard Controls.

Control	Property	Description
CheckBox	HighlightBorderStyle	Border properties for highlighting a CheckBox

	HighlightBorderWidth HighlightBorderColor	<p>when selected by the user.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
CheckBoxList	HighlightBorderStyle HighlightBorderWidth HighlightBorderColor	<p>Border properties for highlighting a CheckBox when selected by the user.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
PushButton	HighlightBorderStyle HighlightBorderWidth HighlightBorderColor	<p>Border properties for highlighting the PushButton within a group that matches the value of the data field returned from the host system.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
PushButtonList	HighlightBorderStyle HighlightBorderWidth HighlightBorderColor	<p>Border properties for highlighting the PushButton within a group that matches the value of the data field returned from the host system.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
RadioButton	HighlightBorderStyle HighlightBorderWidth HighlightBorderColor	<p>Border properties for highlighting a RadioButton when selected by the user.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
RadioButtonList	HighlightBorderStyle	Border properties for highlighting a RadioButton

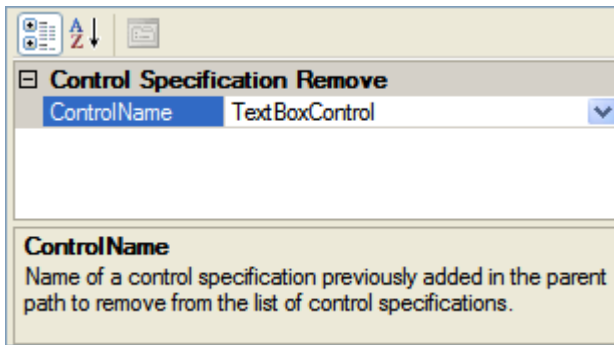
	HighlightBorderWidth HighlightBorderColor	<p>when selected by the user.</p> <p>The property values are identical to those available for Border-Style, -Width, and -Color of ASP.NET controls.</p> <p>When blank or not specified, no highlighting occurs.</p> <p>Default value is "Ridge, 2px".</p>
TextBox	NumericErrorPosition NumericErrorText	<p>These properties have effect only for Textbox controls associated with data fields that are defined as numeric.</p> <p>NumericErrorPosition specifies the position of the error indicator (*) in relation to the TextBox Control. Possible values are: right, left and both. 'both', indicates the error indicator will alternate between left and right. Default value is 'right'.</p> <p>NumericErrorText specifies the error text to show when the user hovers the mouse over the error indicator. The default text is 'Field must be numeric.'</p>
TableRow	AlternatingBackColor	<p>When specified, the background color of every second row in a CopyFrom table is set to this color.</p> <p>The background color of every other row is as specified on the background color property of the Table.</p> <p>When AlternatingBackColor is specified for both TableRow and TableCell a chequered effect is achieved.</p>
TableCell	AlternatingBackColor	<p>When specified, the background color of every second column in a CopyFrom table is set to this color.</p> <p>The background color of every other column is as specified on the background color property of the Table.</p> <p>When AlternatingBackColor is specified for both TableRow and TableCell a chequered effect is achieved.</p>

2.5 Control Specifications, Remove

A Control Specification that has previously been added to a parent in the hierarchy can be removed at any level in the hierarchy by inserting a Control Specification Remove item.

For example, if a specification of a TextBoxControl has been added at the application level, it may be desirable to generate a specific field as a default TextBoxControl. In this case, a Control Specification Remove item can be added to the field to remove the specification for the TextBoxControl.

The properties window of Control Specification Remove is illustrated below.



ControlName: This specifies the name of a Standard Control or Custom Control that previously has been specified at a parent level in the hierarchy. This is selected from the dropdown list, which is automatically filled with a list of available Standard Controls and Custom Controls. For convenience, a 'Remove All' item is available in the list. When selecting 'Remove All', all control specifications inherited from the parent levels will be removed.

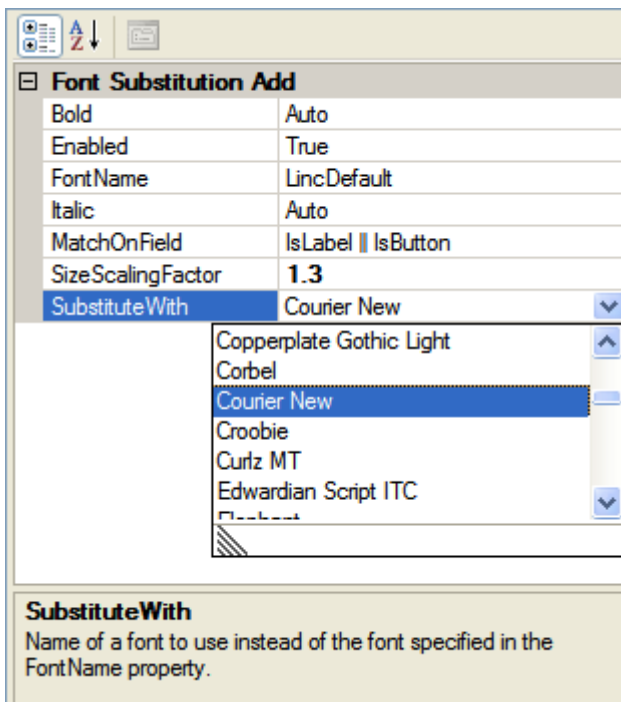
2.6 Font Substitutions, Add

Font Substitutions are used when replacing/substituting fonts that are specified at the time when painting a form with a different font.

Font Substitutions can be added to any level in the hierarchy.

When font substitutions are inherited from parent levels, font substitutions at the current level are evaluated first. When substituting the same font with different fonts specifying match conditions, the substitutions will be evaluated in the order they are specified, except for unconditional substitutions which will be evaluated last.

When adding a Font Substitution, the properties as illustrated below are available.



Bold: This specifies whether to bold the font. The following selections are available:

- Auto – Use the bold setting as specified on the original font.
- On – Bold the font.
- Off – Do not bold the font.

Enabled: This property provides a convenient way of removing a font substitution from the generate process without deleting it.

FontName: This specifies the name of the Font to be substituted. This can be selected from the dropdown list of fonts or typed directly.


Italic: specifies whether to use italic font. The following selections are available:

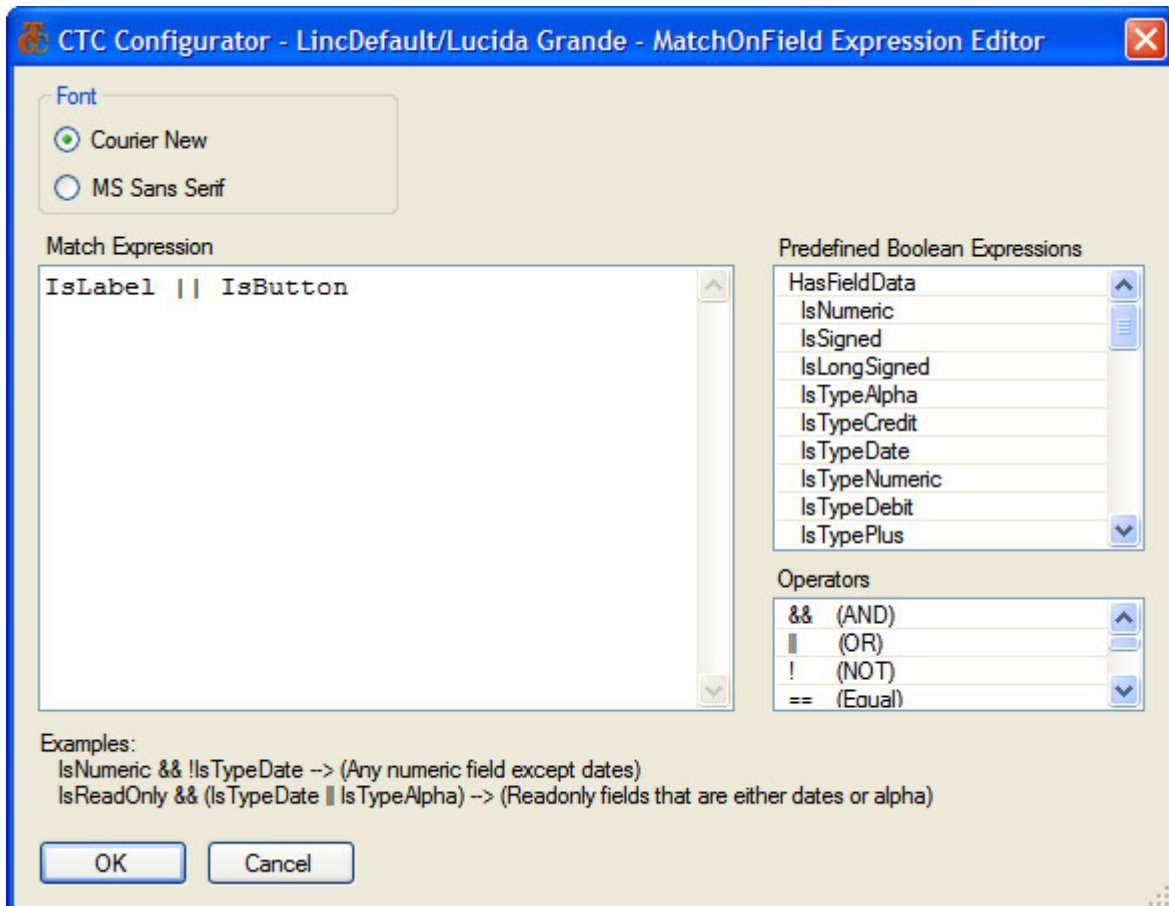
- Auto – Use the italic setting as specified on the original font.
- On – Italic the font.
- Off – Do not italic the font.

SizeScalingFactor: This property specifies the scaling, which is used for increasing/decreasing the font size. When substituting fonts, it may be necessary to increase/decrease the painted font size to get a better match with the substitute font.

SubstituteWith: This specifies the name of a Font to use instead. This can be selected from the dropdown list of fonts or typed directly.

MatchOnField: Optional match expression. MatchOnField is a boolean expression specifying the condition for selecting to which field or fields to apply the substitution. At generate time, the MatchOnField expression is evaluated against the field associated with the particular font and when the expression evaluates true, the substitution is applied. When no expression is specified, control substitution is unconditionally applied.

The expression is specified by selecting the ellipsis button . This will open the Expression Editor window as illustrated below.



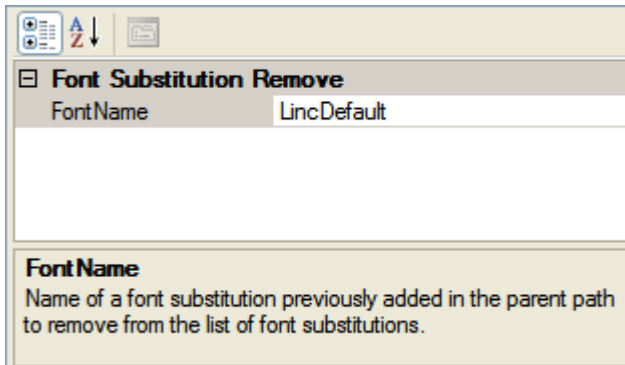
An expression is specified using a selection of predefined boolean expressions. Expressions are dragged/dropped or copied/pasted from the list of Predefined Boolean Expressions. Operators such as && (AND), || (OR), ! (NOT), == (Equal), != (Not Equal), > (Greater Than), < (Less Than), >= (Greater Than or Equal) or <= (Less Than or Equal) as well as parentheses () can be used to create complex expressions.

2.7 Font Substitutions, Remove

A Font Substitution that has previously been added to a parent in the hierarchy can be removed at any level in the hierarchy by inserting a Font Substitution Remove item.

For example, if a substitution of 'Arial' with 'Courier New' has been added at the application level, it may be desirable to generate a specific field as the painted 'Arial' font and not as 'Courier New'. In this case, a Font Substitution Remove item can be added to the field specifying to remove the substitution for the 'Arial' font.

The properties window of Font Substitution Remove is illustrated below.



FontName: This specifies the name of a font that previously has been substituted at a parent level in the hierarchy. This can be selected from the dropdown list of fonts or typed directly. For convenience, a 'Remove All' item is available in the list. When selecting 'Remove All', all font substitutions inherited from the parent levels will be removed.

2.8 Options

Options specify various settings that take effect at generate time.

Options can be specified at any level in the hierarchy.

When selecting an option from the tree structure, for convenience the complete list of options for that level will be shown in the properties window highlighting the option selected. This allows editing any or all of the options in the properties window without having to select each option individually from the tree structure.

A sample options properties window is illustrated below including default values.

Options	
AlternateViewCreate	False
AlternateViewRemove	False
ASPNETAJAXControlToolkitLocation	
ASPNETVersion	ASP.NET 2.0 Standard
BuildGeneratedSolution	True <input type="checkbox"/>
CopyFromAsTable	False
CopyFromAutoColumnHeaders	False
CopyFromMultiLineAsSingle	False
CustomCodeModuleCreate	False
CustomCodeModuleRemove	False
DefaultImageType	jpg
DefaultDateFormat	UK
DisplayErrors	True
DisplayWarnings	True
EnableViewState	True
GenerateSystemInfo	True
IISReset	True
InfrastructureFilesVersionCheck	True
KeepSessionAlive	False
LabelIdDetection	False
LabelIdEndSeparator]
LabelIdStartSeparator	[
ListBox_SubmitOnDoubleClick	False
ListBox_SubmitOnEnterKey	False
LogLevel	3 - Info
ReInstallBundle	False
TextBox_AutoTab	False
TextBox_LabelWhenReadOnly	True
TextBox_MultiLineMaxLengthCheck	True
TextBox_ValidateNumeric	True
TextTransparent	True
ToolTip	False
VirtualDirectoryNameNew	SAMPLE_INQUIRY
VisualStudioVersion	VS2005
XScalingFactor	1
YScalingFactor	1

BuildGeneratedSolution
Automatically compile the generated application and build the required dll's.

The table below lists the options available for the CTC ASP.NET WebForms Generator.

Option	Description
AlternateViewCreate	<p>When set, a copy of the current generated Ispec View is created and added to the generated project for each ispec this option applies to.</p> <p>Creating an alternate view from the generated Ispec View provides an easy way to get started on designing a form using external tools such as Visual Studio WebForm's painter.</p> <p>Once added, this alternate view is not affected in any</p>

	way by the generate unless removed using the AlternateViewRemove option.
AlternateViewRemove	When set, the reference to the alternate view is removed from the generated project for each ispec this option applies to. The alternate view files are not deleted.
ASPNETAJAXControlToolkitLocation	The path where the ASP.NET AJAX Control Toolkit dll is installed. This is required when selecting the ASP.NET 2.0 AJAX version and using any of the ASP.NET AJAX Toolkit controls in the generated solution.
ASPNETVersion	<p>The ASP.NET Version to use when setting up the bundle and generating the user interface application for this bundle. When changing this property, infrastructure files appropriate for the selected option will be reinstalled next time when starting the generate of this bundle.</p> <p>The properties ReInstallBundle and ASPNETVersionChange are set as a result of changing ASPNETVersion.</p>
ASPNETVersionChange	<p>When true, infrastructure files related to the chosen ASP.NET Version will be reinstalled next time when starting the generate of this bundle.</p> <p>This property is read-only and cannot be specified directly.</p>
BuildGeneratedSolution	Automatically compile the generated application and build the required dll's.
CopyFromAsTable	When set, the CopyFrom region painted on a form is generated as a Table with one row for each CopyFrom copy and each control placed within the appropriate cell.
CopyFromAutoColumnHeaders	When set, the generator will look for Label controls painted on the form directly above the CopyFrom region and automatically place them as column headers within the Table.
CopyFromMultiLineAsSingle	When set, the generator will generate a CopyFrom spanning multiple lines as single lines. This option may be useful for MultiLine CopyFrom areas that are not suitable as table layout.
CustomCodeModuleCreate	When set, an empty code-behind module is created and added to the generated project for each ispec form this option applies to. A custom code module allows adding custom code to an ispec form. Once added, this module is not affected in any way by the generator unless specifically removed using the CustomCodeModuleRemove option.

CustomCodeModuleRemove	When set, the reference to the custom code-behind module is removed from the generated project for each ispec form to which this option applies. The custom code module file is not deleted.
DefaultImageType	The file extension to be applied to the image name when the image name returned from the host system doesn't contain an extension.
DefaultDateFormat	The default date format to apply to Date Controls. Possible formats are: <ul style="list-style-type: none"> • UK (ddMMyy, ddMMyyyy) • US (MMddy, MMddyyy) • International (yyMMdd, yyyyMMdd)
DisplayErrors	Display errors occurring during the generate process in a message box.
DisplayWarnings	Display warnings occurring during the generate process in a message box.
EnableViewState	Indicates whether controls automatically save their state in a hidden field during round-trips between the browser and the web server at runtime. Typically, EAE and AB Suite systems do not benefit from saving view state between transactions as a new form is returned from the EAE and AB Suite host system for every transaction. Setting this to false will reduce the amount of data sent to and from the browser.
GenerateSystemInfo	Writes information about ispecs and fields contained in the bundle being generated to the CTCSystemInfo.xml file. This information is used by the configurator to populate the dropdowns on fields, ispecs, bundles and applications with valid selections. When disabled, the information for the current bundle is removed from the xml file on the next generate.
IdentifyGroupBox	Identifies Rectangles with Labels overlapping the top line of the rectangle as a GroupBox. This allows using the IsGroupBox expression in a MatchOnField expression for identifying rectangles and labels painted as a GroupBox.
IISReset	Resets (Stop and Start) IIS before generating Ispec Models files. This avoids errors occurring during the generation of the Ispec Model files caused by IIS locking Ispec Models files that has previously been accessed.
InfrastructureFilesVersionCheck	Enables version check on infrastructure files. When

	<p>true, version check is performed when copying infrastructure files when the ReInstallBundle option is set. Only infrastructure files that are newer than the destination files will be copied.</p> <p>When false, no version check is performed and all infrastructure files are unconditionally copied and existing destination files will be overwritten.</p> <p>Upon successful completion of the generate, the option is automatically reset to true by the generator.</p>
KeepSessionAlive	Ensure inactive sessions don't timeout by sending a background request to the web server 30 seconds before the session is about to time out.
LabelIdDetection	<p>Detect label identifier specified in the text of the label.</p> <p>Since labels are not associated with data items and therefore they are not individually identified in the painter, this provides a convenient way of identifying labels allowing for configuration of specific label controls on the form.</p> <p>For example, "[Head1]Customer Maintenance" where 'Head1' is the name identifying the label allowing this label control to be referred to as Head1 when configuring the label.</p> <p>Label identifier must be alphanumeric containing at least one alpha character or _ (underline).</p> <p>Label identifiers are automatically removed from the label text.</p>
LabelIdEndSeparator	End separator character for identifying labels.
LabelIdStartSeparator	Start separator character for identifying labels. Any text between Start Separator and End Separator within label text identifies a label.
ListBox_SubmitOnDoubleClick	Allow submitting the form to the server when double click on an item in a listbox.
ListBox_SubmitOnEnterKey	Allow submitting the form to the server when hitting the enter key on an item in a listbox.
LogLevel	Level of details to write to the log file.
PositionLeftAdjustment	This specifies a number of pixels to adjust (+/-) the left position of controls. When specified, the left position of controls this option applies to will be modified with the value, which results in the controls being moved left or right in the horizontal direction.
PositionTopAdjustment	This specifies a number of pixels to adjust (+/-) the top position of controls. When specified, the top position of controls this option applies to will be modified with the value, which results in the controls being moved up or down in the vertical direction.

ReInstallBundle	Re-install the bundle by copying infrastructure files to the bundle views directory. This is required when new infrastructure files have been added or to repair damaged files. The re-install is performed when next starting the generate of this bundle. Upon successful completion of the generate, the option is automatically reset by the generator.
RemoveButtonGroupPanel	Removes the panel which is added by AB Suite as a group container for Button Groups when importing a model from EAE.
TextBox_AutoTab	Automatically tab to the next control in the tab order sequence when the maximum number of characters defined for the TextBox has been entered.
TextBox_LabelWhenReadOnly	Indicates whether to generate a Label control instead of a TextBox when the data field is read-only and defined as usage inquiry.
TextBox_CheckMultiLineMaxLength	Ensure the number of characters entered into a multi line textbox doesn't exceed the maximum number of characters defined for the field.
TextBox_ValidateNumeric	Ensure the value entered into a textbox is numeric for fields defined as numeric.
TextTransparent	Ensure controls such as Label and Panel are generated with transparent background.
ToolTip	Display the Help Text, when defined for the field, as a tool tip on the control.
VirtualDirectoryAutoCreate	This option configures the generator to automatically create a virtual directory for the generated Silverlight application. On machines without IIS, this option should set to false.
VirtualDirectoryNameCurrent	The current name of the virtual directory associated with this bundle. This property is read-only and cannot be specified directly.
VirtualDirectoryNameNew	The new name of the virtual directory to be associated with this bundle. The name of the virtual directory will be changed on the next generate.
VisualStudioVersion	The Visual Studio Version allows choosing the version of Visual Studio (VS2005 or VS2008) to use when setting up the bundle and generating the user interface application for this bundle. When changing this property, infrastructure files appropriate for the selected option will be reinstalled next time when starting the generate of this bundle. The properties ReInstallBundle and VisualStudioVersionChange are set as a result of

	changing VisualStudioVersion.
VisualStudioVersionChange	When true, infrastructure files related to the chosen Visual Studio Version will be reinstalled next time when starting the generate of this bundle. This property is read-only and cannot be specified directly.
XScalingFactor	X-axis scaling factor to apply for increasing/reducing the left position and width of controls.
YScalingFactor	Y-axis scaling factor to apply for increasing/reducing the top position and height of controls.

2.9 Runtime Configuration

Runtime Configuration specifies parameters to take effect at run time for the generated user interface application.

Runtime Configuration can be specified at bundle level only.

In order for the generator to update the Web.config file only when the configuration parameters have changed, the `_UpdateWebConfig` parameter is automatically set to true when a parameter is changed. However, `_UpdateWebConfig` can be set false in case it is not convenient to update the Web.config file on the next generate.

At generate time, the Runtime Configuration parameters will be written to the `<appSettings>` section in the Web.config file when the `_UpdateWebConfig` option is set. Any existing parameters will be overwritten.

`_UpdateWebConfig` parameter will automatically be set false by the generator when the Web.config file has been updated. Because the generator updates the configuration file, it is recommended to close the CTC Configurator while running the generator.

A sample Runtime Configuration properties window is illustrated below.

RuntimeConfiguration	
_UpdateWebConfig	False
AppDirRelativePath	
ApplicationName	SAMPLE
ApplicationSwitching	False
AutoPopupErrorWindow	True
BundleName	TESTOO
CEWFRLoggingEnabled	False
ConfigName	DefaultSystem
ConfigPrefix	
ConnectionMode	0
DefaultImageType	.jpg
DisplayName	SAMPLE
EnableMessageQueuing	False
FireUpIspecForPooling	MENU
HostDomain	.
HostLogin	ALPublic
HostPassword	ALPublic
HostURI	x-ratl:localhost:4323
HostViewName	SAMPLE
IspecModelRootDirectory	[IspecModelRootDirectory]
KeepSessionAlive	False
LINCEnvironmentProgId	
LogFile	C:\Temp\ASPNet.log
LoggingEnabled	False
LogLevel	7
MultiSessionLogging	False
ObjectPoolingAssembly	
ObjectPoolingEnabled	False
ObjectPoolingType	
PackagePrefix	[PackagePrefix]
PreserveWhiteSpace	True
RATLStartVersion	13
ReverseMsgSequence	False
StartUpIspec	
UseLoginForm	False

_UpdateWebConfig
Update the Web.config file with the runtime configuration parameters. Updating the Web.config file is performed when next starting the generate of this bundle. Upon successful completion of the generate, the option is automatically reset by the generator.

For information on the Runtime Configuration parameters, refer to the relevant section in the documentation for Component Enabler.