

CTC WCF Services Configurator

Version 2.0.0



Table of Contents

| | | |
|-------|---|----|
| 1 | Introduction | 3 |
| 1.1 | What is CTC WCF Services Configurator?..... | 3 |
| 1.2 | The Concept | 3 |
| 1.3 | Configurable Elements | 3 |
| 1.4 | License Keys | 4 |
| 1.5 | Standard Data Type Controls..... | 4 |
| 2 | Configuration Items | 5 |
| 2.1 | Custom Data Type Controls..... | 5 |
| 2.2 | Control Substitutions, Add..... | 6 |
| 2.3 | Control Substitutions, Remove..... | 7 |
| 2.4 | Control Specifications, Add..... | 8 |
| 2.4.1 | Data Type Control Properties | 10 |
| 2.5 | Control Specifications, Remove..... | 12 |
| 2.6 | Options | 13 |
| 2.7 | Runtime Configuration | 21 |

1 Introduction

1.1 What is CTC WCF Services Configurator?

The CTC WCF Services Configurator is the Configurator for the WCF Services 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 WCF Services Generator.

This document should be read in conjunction with the **CTC Configurator Framework** document and the **CTC WCF Services Generator** document.

1.2 The Concept

Typically, a generator creates a fixed, pre-determined solution, where users have limited or no influence on what is generated. Customizations have to be applied by modifying the generated solution, 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 solution after it has been generated.

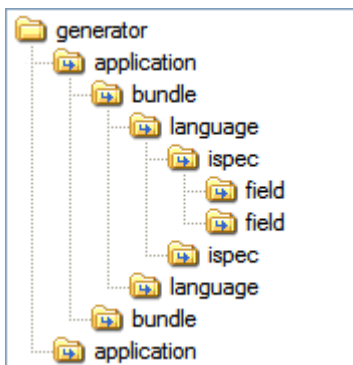
This requires the generator to be very flexible and to provide the means for customizing the result of the generate prior to entering the generate phase.

CTC Generators provide the ability to influence what is generated by configuring features, setting options, customizing Standard Data Type Controls, adding Custom Data Type Controls and substituting Controls. The generated solution is still based on the forms and controls being painted in the EAE or AB Suite development environment, however, the CTC WCF Services 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 WCF Services Generator within each of the configurable elements (generator, application, bundle, language, ispec and field):

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

1.4 License Keys

License Keys are required for using the generator and can be specified at the generator level only.

License keys for the WCF Services 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 WCF Services Generator for the first time 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 WCF Services Generator.

When sharing configuration files between work stations 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 Data Type Controls

Standard Data Type Controls provide the default implementation of the data items/fields data type as they are specified in the EAE and AB Suite Development environments. Standard Data Type Controls are built into the WCF Services Generator. They can be used as they are without further customization or they can be customized.

The following is the list of Standard Data Type Controls available with the WCF Services Generator:

| Control | Description |
|--------------|---|
| AlphaType | Controls how fields defined as alpha-numeric are generated. This includes Alpha and String types. |
| BooleanType | Controls how fields painted as single Push Button or single CheckBox that contain true/false values defined at design time are generated. |
| CopyFromType | Controls how CopyFrom data items are generated. |

| | |
|-------------|---|
| DateType | Controls how fields defined as Date are generated. |
| DecimalType | Controls how fields defined as numeric with decimals are generated. This includes Number, Signed, Credit, + and - types. |
| EnumType | Controls how fields painted as a control that contains a list of values defined at design time are generated. This includes push button, radio button, list box and combo box. |
| KanjiType | Controls how fields defined as Kanji are generated. |
| ListType | Controls how fields painted as a list box or combo box creating dynamic lists at runtime are generated. |
| MaintType | Controls how the Maint field of standard component ispecs is generated. |
| NumericType | Controls how fields defined as numeric without decimals are generated. This includes Number, Signed, Credit, + and - types. |
| WideType | Controls how fields defined as Wide are generated. |

2 Configuration Items

2.1 Custom Data Type Controls

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

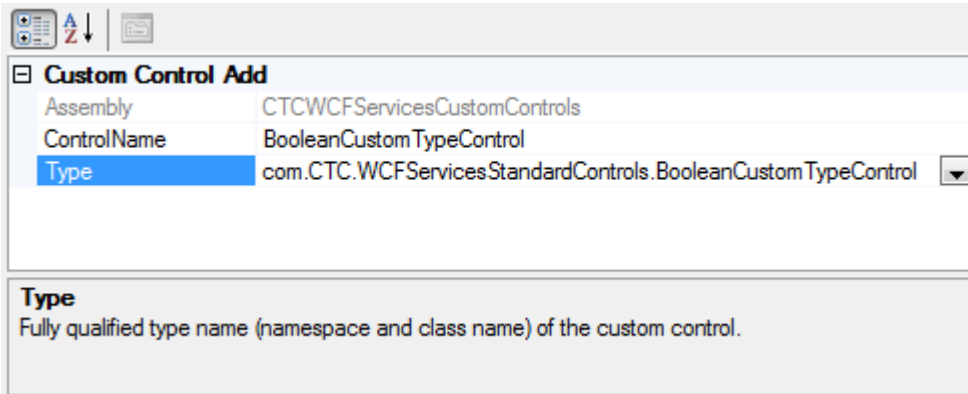
As an example, delivered with the WCF Services Generator is a Boolean Custom Type Control which will convert the value of an ispec field to/from a Boolean true/false. When required, this Custom Data Type Control can substitute the Standard Alpha Type Control for specific fields and expose a field value as a Boolean to the client.

For information on how to create a Custom Data Type Control, refer to the **CTC WCF Services Generator** documentation.

A Custom Data Type 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 Data Type 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 Data Type Control, the ControlName and Type properties as illustrated below must be specified.



| Custom Control Add | |
|--------------------|--|
| Assembly | CTCWCFServicesCustomControls |
| ControlName | BooleanCustomTypeControl |
| Type | com.CTC.WCFServicesStandardControls.BooleanCustomTypeControl ▼ |

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

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

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

Type: This is selected from the dropdown list, which is automatically filled with a list of Custom Data Type 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 Data Type Control.

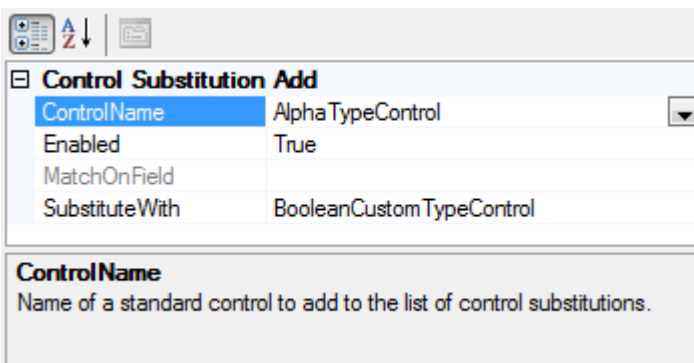
2.2 Control Substitutions, Add

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

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

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 Data Type 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 | AlphaTypeControl ▼ |
| Enabled | True |
| MatchOnField | |
| SubstituteWith | BooleanCustomTypeControl |

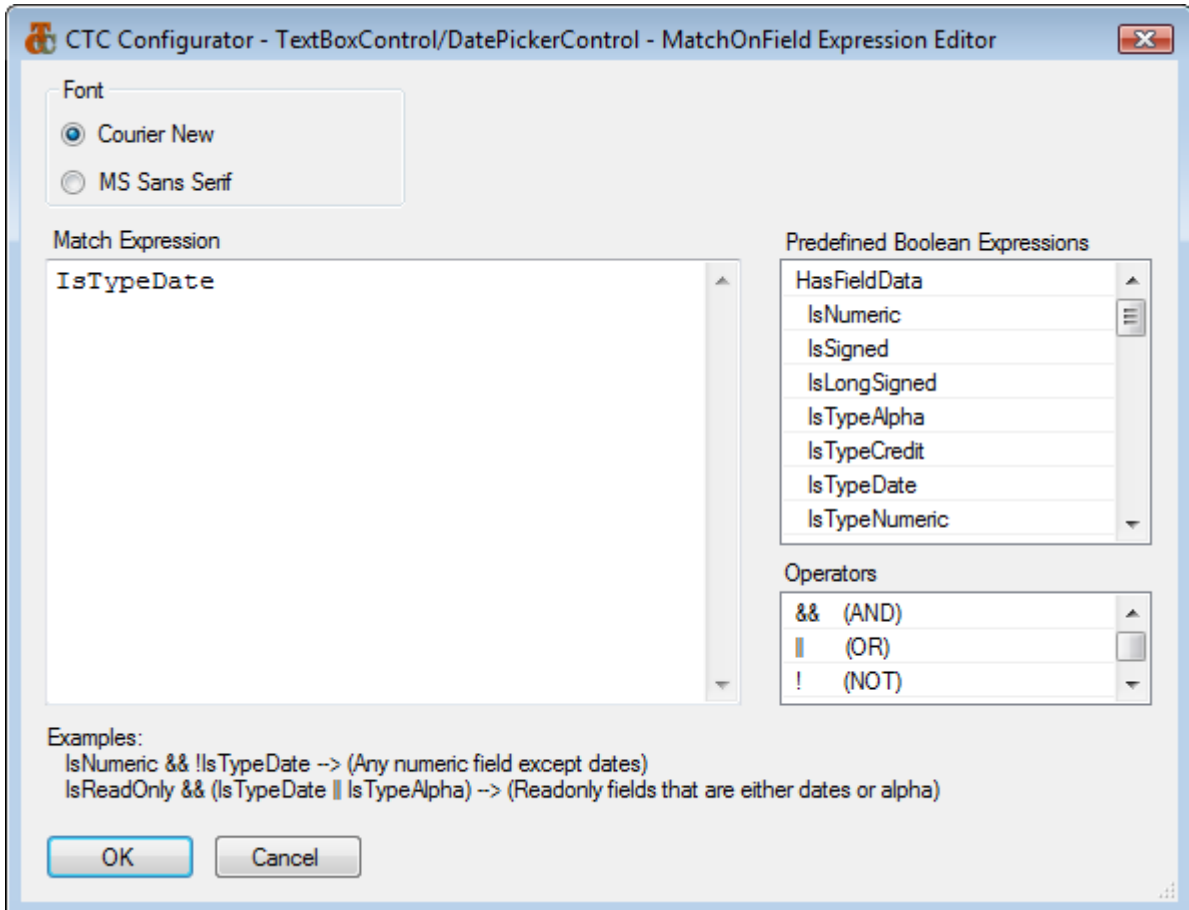
ControlName
Name of a standard control to add to the list of control substitutions.

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

Enabled: This property provides a convenient way of removing a control substitution 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 . 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.

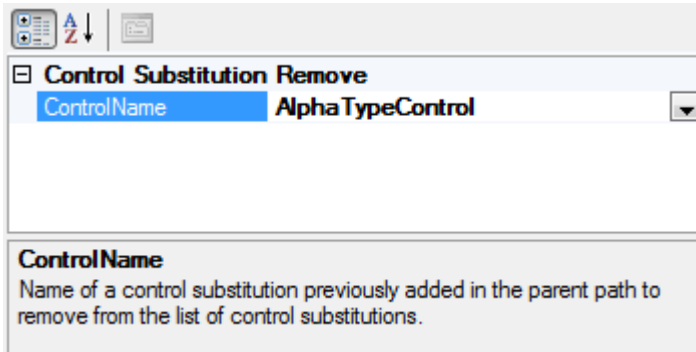
SubstituteWith: This specifies the name of a Custom Data Type Control to use instead of the Standard Data Type Control. This is selected from the dropdown list, which is automatically filled with a list of Custom Data Type Controls currently added to the generator.

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

The properties window of Control Substitution Remove is illustrated below.



| Control Substitution Remove | |
|-----------------------------|------------------|
| ControlName | AlphaTypeControl |

ControlName
Name of a control substitution previously added in the parent path to remove from the list of control substitutions.

ControlName: This specifies the name of a Standard Data Type 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 Data Type 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

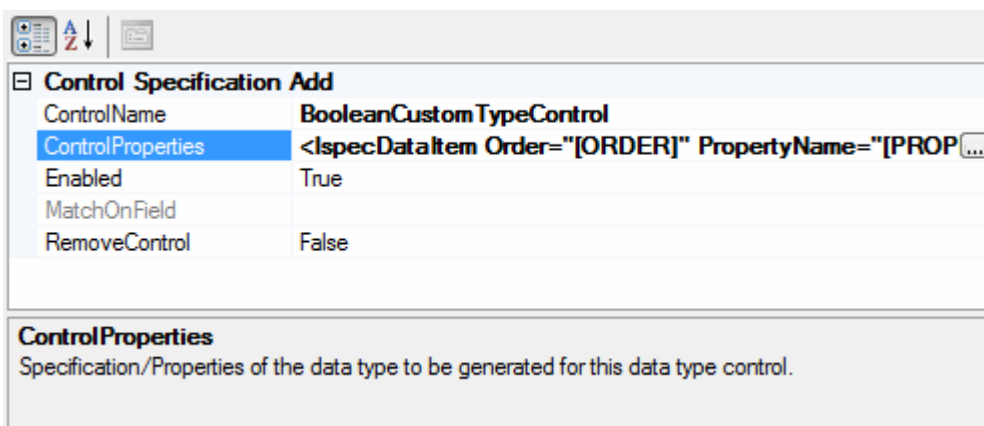
Control Specifications are used to change the default specifications of Standard Data Type Controls and Custom Data Type Controls. This allows modification of the default template of a control.

The specification of a control specifies how the data type is generated.

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.

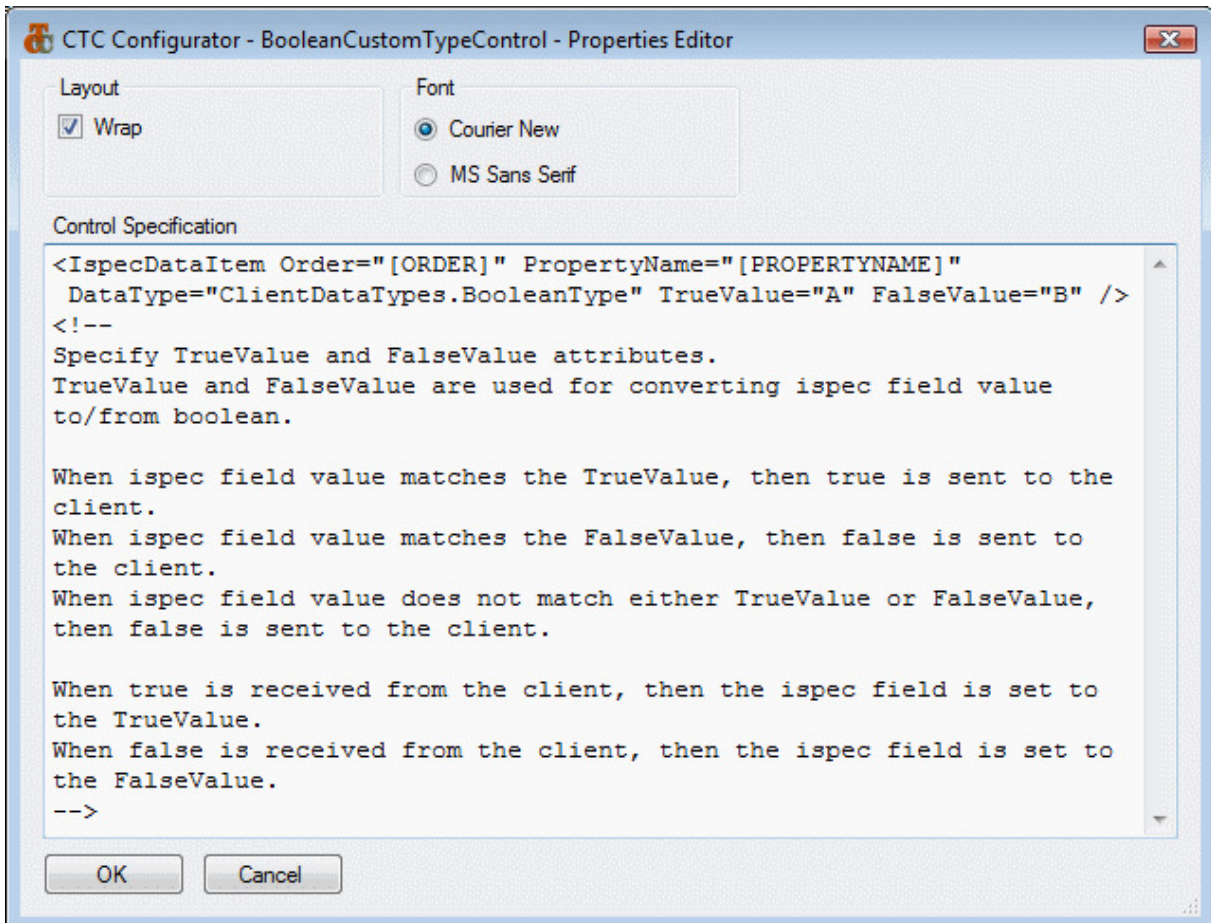


| Control Specification Add | |
|---------------------------|--|
| ControlName | BooleanCustomTypeControl |
| ControlProperties | <!specDataItem Order="[ORDER]" PropertyName="[PROP...] |
| Enabled | True |
| MatchOnField | |
| RemoveControl | False |

ControlProperties
Specification/Properties of the data type to be generated for this data type control.

ControlName: This specifies the name of a Standard Data Type Control or a Custom Data Type Control for which to specify properties. This is selected from the dropdown list, which is automatically filled with a list of available Standard Data Type Controls and Custom Data Type 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 defines how the data type is generated. Via this editor, the user specifies the properties and their value to be set for the data type.

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.

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.

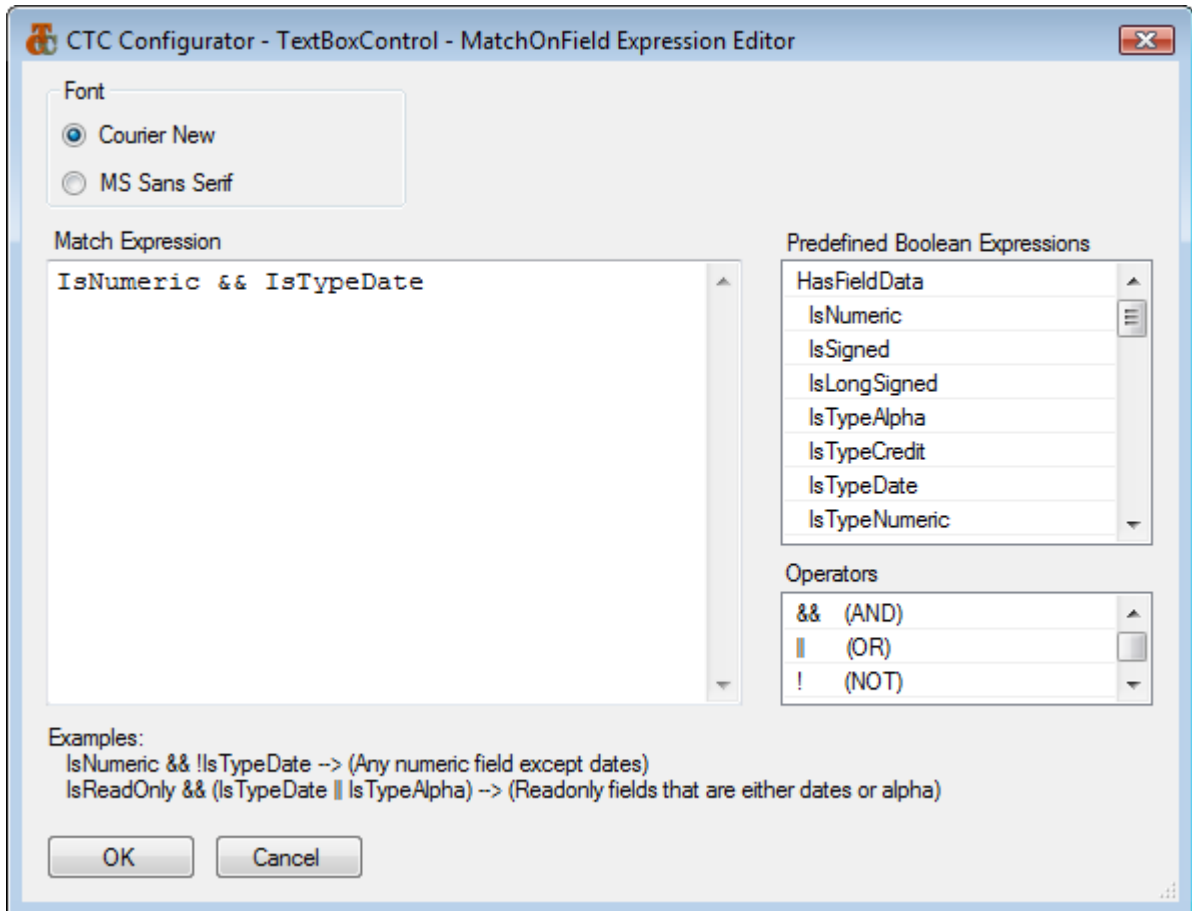
The variable names are case sensitive.

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 data type control is to be removed from the generated Service Model. This is a convenient way of excluding a field from being generated.

2.4.1 Data Type Control Properties

The following properties are available on the data type controls.

| Control | Property | Description |
|----------------|----------|---|
| All Data Types | Order | Specifies the sequence in which a data item will appear in the request/response message. By default the order is determined by the Component Enabler Generate Environment. |

| | | |
|----------------|----------------|---|
| All Data Types | PropertyName | The property name of the data item in the generated Service Model. The default property name is the ispec field name. |
| All Data Types | DataType | Specifies the data type of the data item as it will appear to the client. |
| Boolean Type | TrueValue | Specifies the ispec field value to correspond to the client side true value. |
| Boolean Type | FalseValue | Specifies the ispec field value to correspond to the client side false value. |
| Enum Type | Key | When customizing enumerations to be sent to the client, the Key property represents the value on the host system. One or more Key/Value pairs can be specified. |
| Enum Type | Value | When customizing enumerations to be sent to the client, the Value property represents the value on the client side. One or more Key/Value pairs can be specified. |
| Enum Type | Default | Identifies the default enumeration entry. |
| List Type | XmlFilePath | XmlFilePath specifies an external XML formatted file to send to the client as a list. When specified, the file will automatically be loaded and included in the response to the client. The path to the file is relative to the 'views' directory of the generated bundle. For example: XmlFilePath="ExternalLists/Countries.xml" points to views\ExternalLists\Countries.xml |
| List Type | XmlElementPath | XmlElementPath is an XPath expression specifying the path to the xml element within the file holding the data to include in the list. The path must be specified relative to the root element. For example: XmlElementPath="//country" specifies the country element within the root. The XmlElementPath can specify a complex expression for the path to the element. For example: XmlElementPath="//country[@code='GB']/city" specifies the city element within the country which has the code attribute equal to GB. |
| List Type | PropertyName | Specifies the property name of a list column as it is to appear on the client side. One or more PropertyName/DataType/Column entries can be specified. |

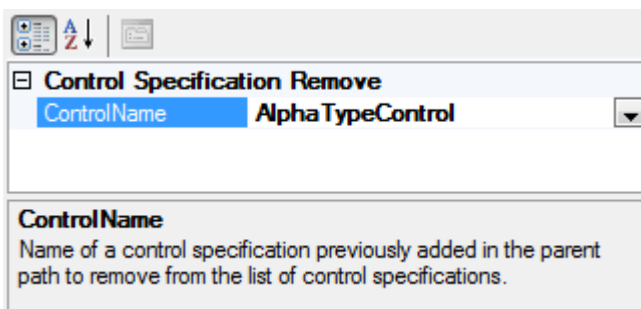
| | | |
|-----------|-----------------|--|
| List Type | DataType | Specifies the data type of a list column as it is to appear on the client side. One or more PropertyName/DataType/Column entries can be specified. |
| List Type | Column | Specifies the column name of a list column from the host system to include in the response message. Columns on a host list are named Key, Column1,...,ColumnN The Column property allows specifying substring. I.e. Column="2(0-10)" specifies to take 10 characters from column 2 of the list from the host starting at index 0. Specifying sub columns provides a way to create a multi column list even when the host system returns a single column list without making any changes to the host system. One or more PropertyName/DataType/Column entries can be specified. |
| List Type | ReturnEmptyList | When true, an empty list is returned when the list does not exist. When false, an error is returned when the list does not exist. Default value is false. |

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

The properties window of Control Specification Remove is illustrated below.



ControlName: This specifies the name of a Standard Data Type Control or Custom Data Type 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

Data Type Controls and Custom Data Type 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 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.

Allow Maint Operations

| | |
|---------------|-------|
| AllowMaintADD | False |
| AllowMaintBAC | False |
| AllowMaintCHG | False |
| AllowMaintDEL | False |
| AllowMaintFIR | False |
| AllowMaintINQ | True |
| AllowMaintLAS | False |
| AllowMaintNEX | False |
| AllowMaintPUR | False |
| AllowMaintREC | False |

Misc Options

| | |
|---------------------------------|-------------------------------|
| AlternateServiceCreate | False |
| AlternateServiceOriginalRemove | False |
| AlternateServiceRemove | False |
| BuildGeneratedSolution | True |
| CreateWebServicesSolution | True |
| CustomCodeModuleCreate | False |
| CustomCodeModuleRemove | False |
| Data Type Decimal | Double |
| Data Type Numeric | Integer |
| Default Date Format | UK |
| DisplayErrors | True |
| DisplayWarnings | True |
| GenerateSystemInfo | True |
| IISReset | True |
| IncludeListDataInResponse | True |
| IncludeUsageInputInResponse | False |
| IncludeUsageInquiryInResponse | True |
| IncludeUsageIOInResponse | True |
| InfrastructureFilesVersionCheck | True |
| ListOfValuesAsEnum | False |
| LogLevel | 3 - Info |
| ReInstallBundle | False |
| SingleButtonAsBoolean | False |
| SvcUtilLocation | C:\Program Files\Microsoft SC |
| UseDataDisplayAsName | False |
| UseSpecDescriptionAsName | False |
| VirtualDirectoryNameCurrent | Sample_Testwcf |
| VirtualDirectoryNameNew | SAMPLE_TESTWCF |
| VisualStudioVersion | VS2008 |

AllowMaintADD
Specifies whether to allow the Add (ADD) operation to be generated for Standard Component ispecs.

Naming Operations

| | |
|-------------------|-------------|
| NameMaintADD | Add |
| NameMaintBAC | Prior |
| NameMaintCHG | Change |
| NameMaintDEL | Delete |
| NameMaintFIR | First |
| NameMaintINQ | Inquiry |
| NameMaintLAS | Last |
| NameMaintNEX | Next |
| NameMaintPUR | Purge |
| NameMaintREC | Recall |
| NameMemoOperation | Transaction |

Naming Types

| | |
|-------------------|----------------|
| NameAlternateType | Alt |
| NameCopyFromType | RepeatingGroup |
| NameInputType | Input |
| NameKeyType | Key |
| NameListType | List |
| NameOutputType | Output |
| NameStatusType | Status |

Operations Input Type

| | |
|------------------------|-----------|
| InputTypeMaintADD | Object |
| InputTypeMaintBAC | Parameter |
| InputTypeMaintCHG | Object |
| InputTypeMaintDEL | Parameter |
| InputTypeMaintINQ | Parameter |
| InputTypeMaintNEX | Parameter |
| InputTypeMaintPUR | Parameter |
| InputTypeMaintREC | Parameter |
| InputTypeMemoOperation | Object |

NameMaintADD
Specifies the name of an Add (ADD) operation.

The table below lists the options available for the CTC WCF Services Generator.

| Option | Description |
|-------------------------------|--|
| Allow Maint Operations | |
| AllowMaintADD | Specifies whether to allow the Add (ADD) operation of a Standard Component to be generated. |
| AllowMaintBAC | Specifies whether to allow the Back (BAC) operation of a Standard Component to be generated. |
| AllowMaintCHG | Specifies whether to allow the Change (CHG) operation of a Standard Component to be generated. |
| AllowMaintDEL | Specifies whether to allow the Delete (DEL) operation of a Standard Component to be generated. |

| | |
|--------------------------------|---|
| AllowMaintFIR | Specifies whether to allow the First (FIR) operation of a Standard Component to be generated. |
| AllowMaintINQ | Specifies whether to allow the Inquiry (INQ) operation of a Standard Component to be generated. |
| AllowMaintLAS | Specifies whether to allow the Last (LAS) operation of a Standard Component to be generated. |
| AllowMaintNEX | Specifies whether to allow the Next (NEX) operation of a Standard Component to be generated. |
| AllowMaintPUR | Specifies whether to allow the Purge (PUR) operation of a Standard Component to be generated. |
| AllowMaintREC | Specifies whether to allow the Recall (REC) operation of a Standard Component to be generated. |
| Misc Options | |
| AlternateServiceCreate | <p>When set, a copy of the current generated Service Model is created and added to the generated project for each ispec this option applies to.</p> <p>Creating an alternate service from the generated Service Model provides an easy starting point for creating a custom Service Model.</p> <p>Once added, this alternate service is not affected in any way by the generate unless removed using the AlternateServiceRemove option.</p> |
| AlternateServiceOriginalRemove | When set and an Alternate Service exist for an ispec, the original Ispec ServiceModel will be removed from the generated project. Removing the original Ispec ServiceModel from the project keeps the size of the Adapter dll to a minimum. |
| AlternateServiceRemove | When set, the reference to the alternate service is removed from the generated project for each ispec this option applies to. The alternate service file is not deleted. |
| BuildGeneratedSolution | Automatically compile the generated solution and build the required dll's. |
| CreateWebServicesSolution | Automatically create a Web Services solution for all operations included in the generated WCF Adapter solution. This can be used for testing the Adapter services using the WCF Test Client. |
| CustomCodeModuleCreate | <p>When set, an empty code-behind module is created and added to the generated project for each ispec this option applies to. A custom code module allows the addition of custom code to an ispec service model.</p> <p>Once added, this module is not affected in any way by the generator unless specifically removed using the CustomCodeModuleRemove option.</p> |

| | |
|-----------------------------|--|
| CustomCodeModuleRemove | When set, the reference to the custom code-behind module is removed from the generated project for each ispec this option applies. The custom code module file is not deleted. |
| DataTypeDecimal | Specifies the client side type to represent fields defined as numeric with decimals. This applies to Number, Signed, Credit, + and –. The possible values are: Decimal or Double. |
| DataTypeNumeric | Specifies the client side type to represent fields defined as numeric without decimals. This applies to Number, Signed, Credit, + and –. The possible values are: Integer or Long. |
| DefaultDateFormat | Specifies the default date format of date fields used when converting dates to/from standard client side DateTime format. Possible formats are: <ul style="list-style-type: none"> - UK (ddMMyy, ddMMyyyy) - US (MMddy, MMdyyyy) - 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. |
| 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. |
| IISReset | Resets (Stop and Start) IIS before generating Ispec Model files. This avoids errors occurring during the generation of the Ispec Model files caused by IIS locking Ispec Model files that have previously been accessed. This option is only applicable when CreateWebServicesSolution option is enabled. |
| IncludeListDataInResponse | Specifies whether to include a list of data in the response message for fields represented as ListBox or ComboBox that create a list of data at runtime. |
| IncludeUsageInputInResponse | Specifies whether to include fields defined as Usage |

| | |
|---------------------------------|--|
| | Input in the response message. |
| IncludeUsageInquiryInResponse | Specifies whether to include fields defined as Usage Inquiry in the response message. |
| IncludeUsageIOInResponse | Specifies whether to include fields defined as Usage IO in the response message. |
| InfrastructureFilesVersionCheck | <p>Enables version check on infrastructure files. When true, version check is performed while 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> |
| ListOfValuesAsEnum | <p>Specifies whether to create an Enum type for fields represented by controls such as PushButton, RadioButton, ListBox and ComboBox that include a list of values.</p> <p>When enabled, the list of values specified at design time for these controls will be generated as an Enum type allowing the client to only select from the Enum list.</p> |
| LogLevel | Level of details to write to the log file. |
| 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. |
| SingleButtonAsBoolean | <p>Specifies whether to create a Boolean type for fields represented by a single control such as PushButton and CheckBox that include true/false values.</p> <p>For PushButton, the action value represents the true value. The false value for a PushButton is always blank.</p> <p>For CheckBox, the Checked value represents the true value and the Unchecked value represents the false value.</p> |
| SvcUtilLocation | Specifies the path to the folder of the SvcUtil used by the generator when automatically creating the Web Services solution. |
| TwoDigitYearCutoff | This option specifies an integer from 1 to 9999 that represents the cutoff year for interpreting two-digit years as four-digit years. This option is used when |

| | |
|-----------------------------|---|
| | <p>presenting the two digit year of 6 digit date fields as a date including the century in controls such as the Silverlight DatePicker.</p> <p>A two-digit year that is less than or equal to the last two digits of the cutoff year is in the same century as the cutoff year. A two-digit year that is greater than the last two digits of the cutoff year is in the century that precedes the cutoff year. For example, if two digit year cutoff is 2056 (the default), the two-digit year 56 is interpreted as 2056 and the two-digit year 57 is interpreted as 1957. In other words, a two digit year cutoff of 2056 specifies dates in the 100 years range between 1957 and 2056. This is the equivalent of the Base Year specified on the Business Segment of EAE and AB Suite, which has a default value of 1957.</p> |
| UseDataDisplayAsName | <p>Specifies whether to use the data display text of fields for creating meaningful property names when creating the Service Model.</p> <p>Duplicated values will have a number appended in order to create unique names.</p> |
| UseIspecDescriptionAsName | <p>Specifies whether to use the description of ispecs to create meaningful service names.</p> <p>Duplicated values will have a number appended in order to create unique names.</p> |
| VirtualDirectoryAutoCreate | <p>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.</p> |
| VirtualDirectoryNameCurrent | <p>The current name of the virtual directory associated with the Web Services solution.</p> |
| VirtualDirectoryNameNew | <p>The new name of the virtual directory to be associated with the Web Services solution.</p> <p>The name of the virtual directory will be changed on the next generate.</p> |
| VisualStudioVersion | <p>The Visual Studio Version is VS2008 and cannot currently be changed.</p> |
| Naming Operations | |
| NameMaintADD | <p>Specifies the name of an Add (ADD) operation.</p> |
| NameMaintBAC | <p>Specifies the name of a Back (BAC) operation.</p> |
| NameMaintCHG | <p>Specifies the name of a Change (CHG) operation.</p> |
| NameMaintDEL | <p>Specifies the name of a Delete (DEL) operation.</p> |
| NameMaintFIR | <p>Specifies the name of a First (FIR) operation.</p> |
| NameMaintINQ | <p>Specifies the name of an Inquiry (INQ) operation.</p> |

| | |
|------------------------------|---|
| NameMaintLAS | Specifies the name of a Last (LAS) operation. |
| NameMaintNEX | Specifies the name of a Next (NEX) operation. |
| NameMaintPUR | Specifies the name of a Purge (PUR) operation. |
| NameMaintREC | Specifies the name of a Recall (REC) operation. |
| NameMemoOperation | Specifies the name of a Memo operation. |
| Naming Types | |
| NameAlternateType | Specifies the suffix name of an Alternate type that is appended to the ispec name in order to create a unique type name. |
| NameCopyFromType | Specifies the suffix name of a CopyFrom type that is appended to the ispec name in order to create a unique type name. |
| NameInputType | Specifies the suffix name of an Input type that is appended to the ispec name in order to create a unique type name. An Input type is created for operations such as Add, Change and Memo when the Operations Input Type is Object. |
| NameKeyType | Specifies the suffix name of a Key type that is appended to the ispec name in order to create a unique type name. A Key type is created for Maint operations such as Back, Delete, Inquiry, Next, Purge and Recall that take keys as input when the Operations Input Type is Object. |
| NameListType | Specifies the suffix name of a List type that is appended to the field name in order to create a unique type name. |
| NameOutputType | Specifies the suffix name of an Output type that is appended to the ispec name in order to create a unique type name. An Output type is created for operations such as Back, First, Inquiry, Last, Next, Recall and Memo. |
| NameStatusType | Specifies the suffix name of a Status type that is appended to the ispec name in order to create a unique type name. A Status type is created for Maint operations such as Add, Change, Delete and Purge that only returns status information from the transaction. |
| Operations Input Type | |
| InputTypeMaintADD | Specifies the type of input for an Add (ADD) operation. Operation input can be generated as: |

| | |
|-------------------|--|
| | <ul style="list-style-type: none"> - Object: Creates an input object named [IspecName]_Input containing the data items that are required as input to the operation. - Parameter: Creates the data items required as input as parameters on the operation. |
| InputTypeMaintBAC | <p>Specifies the type of input for a Back (BAC) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named [IspecName]_Key containing the key data items that are required as input to the operation. - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMaintCHG | <p>Specifies the type of input for a Change (CHG) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates an input object named [IspecName]_Input containing the data items that are required as input to the operation. - Parameter: Creates the data items required as input as parameters on the operation. |
| InputTypeMaintDEL | <p>Specifies the type of input for a Delete (DEL) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named [IspecName]_Key containing the key data items that are required as input to the operation. - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMaintINQ | <p>Specifies the type of input for an Inquiry (INQ) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named [IspecName]_Key containing the key data items that are required as input to the operation. - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMaintNEX | <p>Specifies the type of input for a Next (NEX) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named |

| | |
|------------------------|--|
| | <p>[IspecName]_Key containing the key data items that are required as input to the operation.</p> <ul style="list-style-type: none"> - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMaintPUR | <p>Specifies the type of input for a Purge (PUR) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named [IspecName]_Key containing the key data items that are required as input to the operation. - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMaintREC | <p>Specifies the type of input for a Recall (REC) operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates a key object named [IspecName]_Key containing the key data items that are required as input to the operation. - Parameter: Creates the key data items required as input as parameters for the operation. |
| InputTypeMemoOperation | <p>Specifies the type of input for a Memo operation.</p> <p>Operation input can be generated as:</p> <ul style="list-style-type: none"> - Object: Creates an input object named [IspecName]_Input containing the data items that are required as input to the operation. - Parameter: Creates the data items required as input as parameters on the operation. |

2.7 Runtime Configuration

Runtime Configuration specifies parameters to take effect at run time for the generated solution.

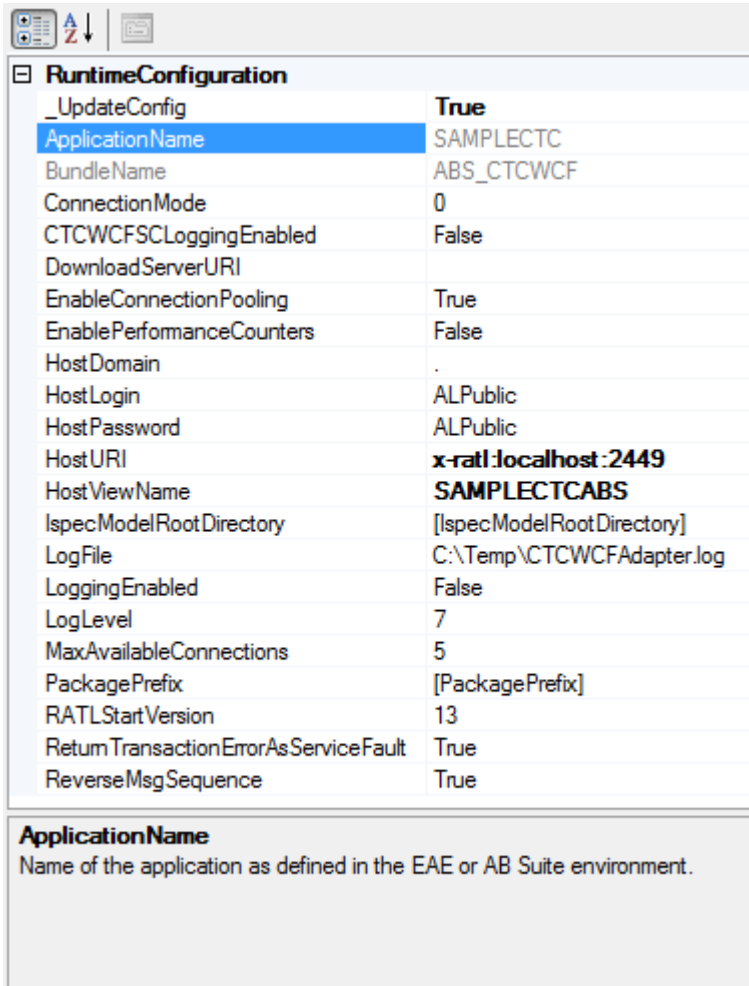
Runtime Configuration can be specified at bundle level only.

In order for the generator to update the Adapter Properties\DefaultConfig.xml file only when the configuration parameters have changed, the `_UpdateConfig` parameter is automatically set to true when a parameter is changed. However, `_UpdateConfig` can be set false in case it is not convenient to update the Adapter Properties\DefaultConfig.xml file on the next generate.

At generate time, the Runtime Configuration parameters will be written to the Adapter Properties\DefaultConfig.xml file when the `_UpdateConfig` option is set. Any existing parameters will be overwritten.

_UpdateConfig parameter will automatically be set false by the generator when the Adapter Properties\DefaultConfig.xml 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.



The table below lists the parameters available.

| Parameter | Description |
|------------------------|--|
| ApplicationName | Name of the application as defined in the EAE or AB Suite environment. |
| BundleName | Name of the bundle/folder as defined in the EAE or AB Suite environment. |
| ConnectionMode | Connection Mode specifies the type of client that is connecting to the host application. |
| CTCWCFSCLoggingEnabled | Enables logging of the CTC WCF Services Controller session to a separate log file. The name '-CTCWCFSC' is appended to the log file name specified in the LogFile parameter. |
| DownloadServerURI | Specifies the server location from where to download IspecModel components. |

| | |
|--------------------------------------|---|
| | <p>DownloadServerURI specifies a virtual directory on a web server. I.e. http://WebServerName/ServerClassesRoot, where 'ServerClassesRoot' specifies a virtual directory on the web server 'WebServerName'.</p> <p>The virtual directory must point to the root location of the IspecModel files (i.e: C:\CE3.3\Classes).</p> <p>When DownloadServerURI is specified, files will be downloaded and cached locally, on the machine on which the Adapter is deployed, to the directory specified on the IspecModelRootDirectory parameter. When a new version of a file is available it will be downloaded.</p> |
| EnableConnectionPooling | Enables Adapter connection pooling. |
| EnablePerformanceCounters | <p>Enables Adapter built-in performance counters.</p> <p>Performance counters can be viewed in the Performance Management Console (perfmon.exe).</p> |
| HostDomain | Host login domain name. |
| HostLogin | Host login userid. |
| HostPassword | Host login password. |
| HostURI | HostURI parameter specifies the connection details according to the required connection type. |
| HostViewName | <p>Host View Name to be used by the Remote Access Server.</p> <p>Note: The View Name is case sensitive.</p> |
| IspecModelRootDirectory | IspecModelRootDirectory parameter specifies the root location of the IspecModel dll's. This is usually the output directory (eg. C:\\CE3.3\\Classes). The variable [IspecModelRootDirectory] will be replaced at generate time with the current value of the output directory. |
| LogFile | File to write logging details to when logging is enabled. |
| LoggingEnabled | Enables logging of the Component Enabler host session. |
| LogLevel | Level of details to write to the log file. |
| MaxAvailableConnections | Specifies the maximum number of available open connections in the Connection Pool at any point in time |
| PackagePrefix | Package Prefix (eg. com.MyCompany) as defined in the EAE or AB Suite environment. The variable [PackagePrefix] will be replaced at generate time with the current package prefix value. |
| RATLStartVersion | RATL Start Version is the initial RATL version that connection should be attempted with. The RATL protocol will automatically fall back to lower versions if connection is unsuccessful, but specifying the value here may make connection to earlier EAE systems quicker. |
| ReturnTransactionErrorAsServiceFault | Specifies how to return transaction errors (CE error code 911) to the Client Application. |

| | |
|--------------------|---|
| | When true (default), a transaction error is returned as a standard Service Fault message. When false, it is returned as a normal response with the error(s) in the TransactionResponseStatus field. |
| ReverseMsgSequence | Specifies the display sequence of error messages returned from the host. |