A Guide to Creating Data Types

Composite 2013-06-21

Contents

compo=ite

1   Introduction

Composite C1 combines page-based and item-based approaches as a content management system. As a page-based system it visually gives you a 1:1 relation between pages created in C1 and the pages on your website. As an item-based system, C1 makes a full use of structured data, the role of which is played by data types in C1, also known as custom content types.
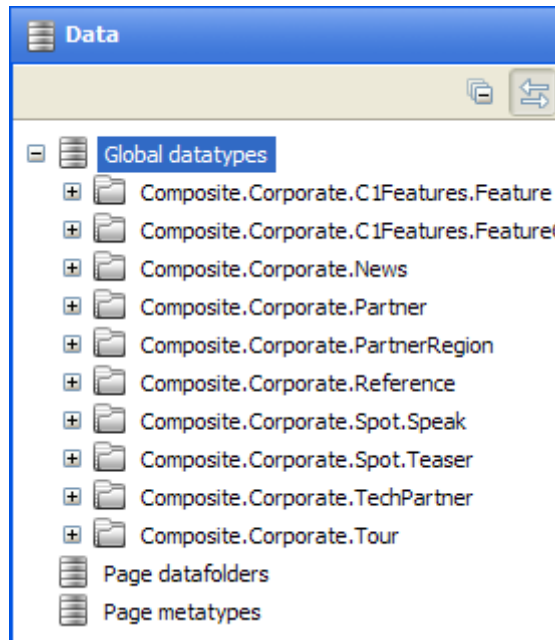

Figure 1: Data types in Data perspective

This guide focuses on creating data types that you can further use on web pages via input forms for entering data. It also gives an overview of data type properties as well as data type fields and widgets used with these fields.

Besides, it features step-by-step instructions on how to create fields that serve specific purposes in input forms. It also provides information about how to add input forms based on data types to web pages by using the C1 Composite.Forms.Renderer package.

1.1   Who Should Read This Guide?

This guide is intended for editors who want to learn to create data types and use them with input forms on web pages.

We assume that you know how to work in the Content perspective of Composite C1. We also assume that you have an idea of structured data such as tables in databases.

As a data type creator, you will primarily work in the Data perspective where you will need permissions to create and edit data types. As a data type user, you will mostly work in the Content perspective where you will need permissions to add and edit pages and data items.

To make use of input forms on the web pages, you will also need to know how to add functions to pages in the Content perspective.

1.2 Getting Started

To get started with data types, you are supposed to take a number of steps.

Getting Started		
Step	**Activity**	**Chapter or section**
1	Create global data types	*2 Creating and Managing Data Types*
2	Add fields to data types	*3 Creating and Managing Fields*
3	Select widgets for data type fields	*4 Widgets*
4	Create specific fields for input forms	*5 Creating Specific Fields*
5	Use input forms with data types	*6 Using Input Forms with Data Types*

In the following few chapters, you will learn more about these and other activities.

1.3 Terms and Abbreviations

The following is the list of terms and their definitions used throughout this guide.

Terms and Definitions	
Term	**Definition**
Data type	An entity used to store and reuse structured data in Composite C1
Global data type	A data type that represents data reusable across all web pages on all web sites created in C1.
Page datafolder	A data type that represents data associated with one specific page.
Page metatype	A data type that represents flexible metadata structures assigned to a page or a section of the website.
Widget	In C1, a control used for getting input values from users and storing them in corresponding data type fields

compo**s**ite

A Guide to Creating Data Types

2 Creating and Managing Data Types

Before you learn how to create data types, you will have a brief overview of different kinds of data types available in Composite C1. You will also find out about general properties of global data types and learn how to add items to these data types and make the data types available for use in the Content perspective.

2.1 Data Types

C1 data types allow you to store structured data and reuse it on your websites.

The data you can reuse in C1 has its scope of visibility. Global data can be accessed and reused across multiple websites and web pages. Page data and metadata is available within one web page.

Accordingly, C1 provides you with three kinds of data types:

- Global data types
- Page datafolders
- Page metatypes

The **Global data types** represent data reusable across all web pages on all web sites created in C1.

The **Page datafolders** represent data associated with one specific page. When created, this data type results in a data folder linked to the page and the editor can store page-specific data in this folder. The data in one page datafolder is not available for other web pages.

The **Page metatypes** represent flexible metadata structures assigned to a page or a section of the website.

The primary focus of this guide is on *global data types*, which allow you to store and reuse data across multiple web pages.
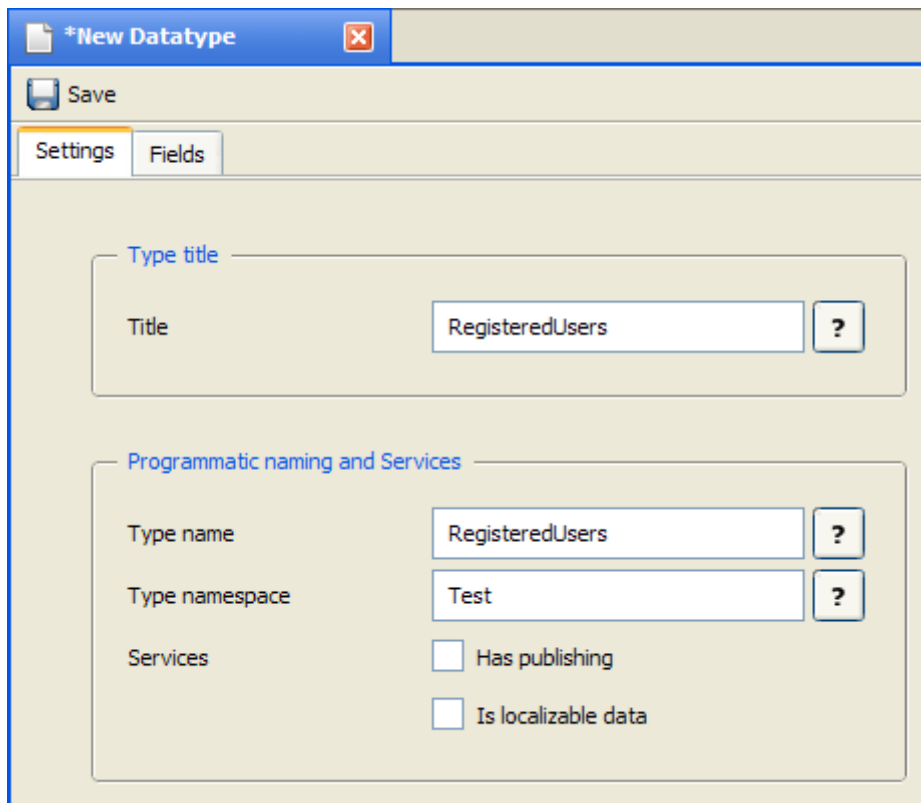
In the following few chapters and sections you will learn how to create global data types, how to set their properties, how to add fields to these data types and how to use forms with data types on web pages.

compo**≡**ite

2.2   Creating Global Data Types

While creating a data type, first, you should set the properties all global data types share. Then, you should add fields to the data type and set properties of each field added.

To create a global data type:

1. In the Data Navigator, select **Global data types** and click **Add Data Type** on the toolbar. The data type settings editor opens in the working area.

2. On the **Settings** tab, set the data type's general properties such as its **Title**, **Type name**, **Type namespace**.

Figure 2: General data type properties

3. On the **Fields** tab, click **Add Field** and on the **Basic** tab of the field settings editor, set the field's basic properties.
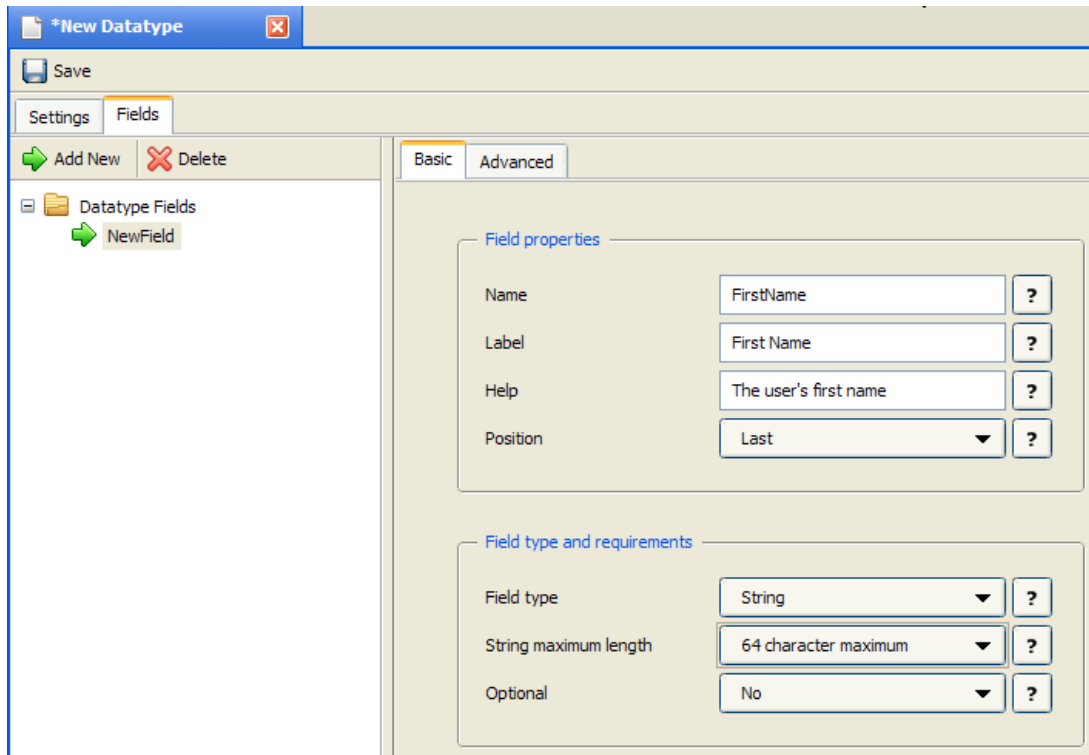

Figure 3: Basic field properties

A Guide to Creating Data Types

4. If necessary, click the **Advanced** tab and set the field's advanced properties.
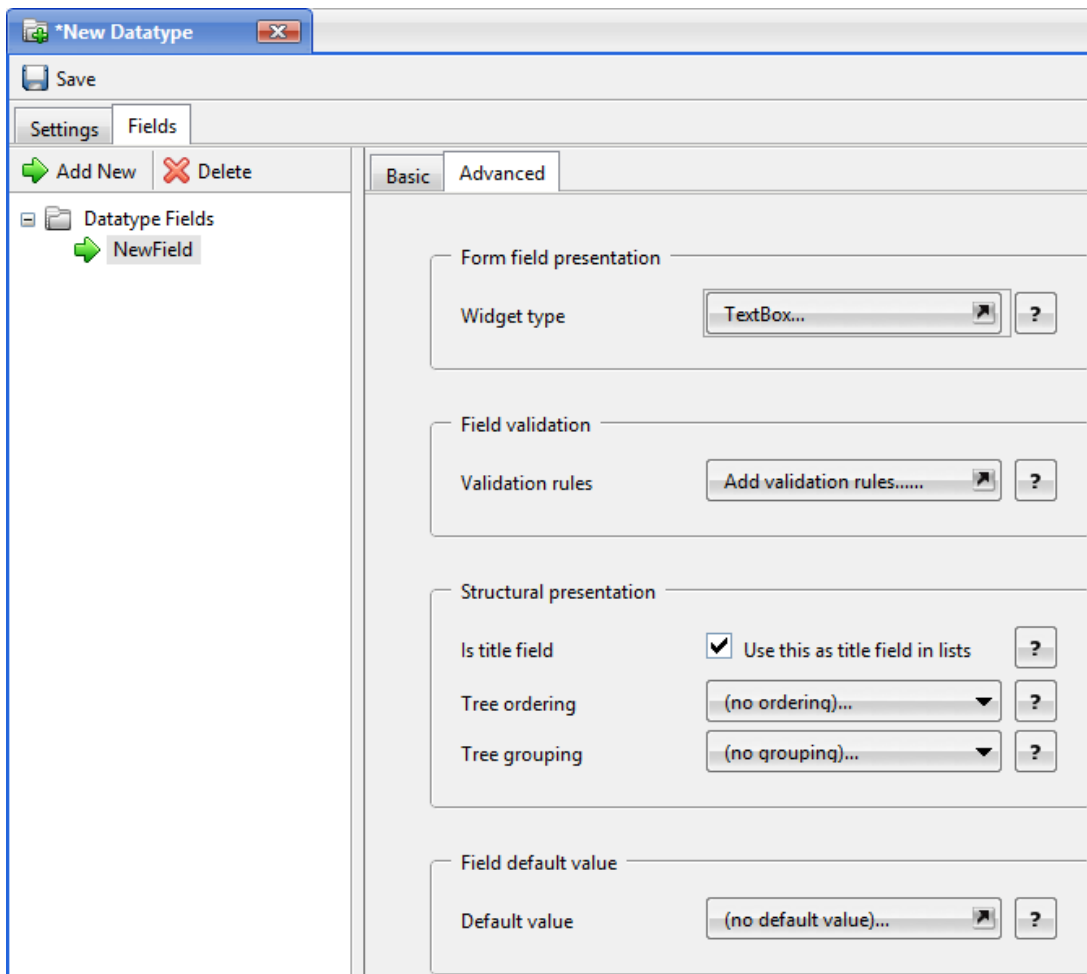

Figure 4: Advanced field properties

5. Repeat *Steps 3* and *4* for as many fields as you need to add.

6. Save the data type.

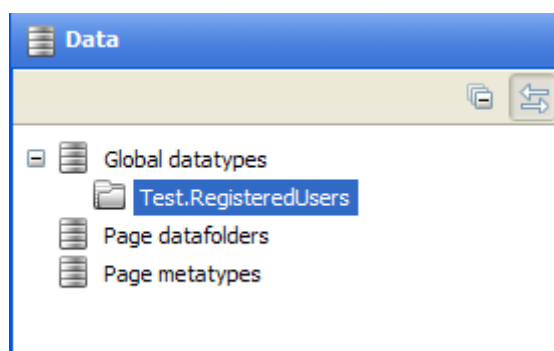The data type appears under Global data types in the Data Navigator.

Figure 5: New global data type

A Guide to Creating Data Types

2.3   General Properties of Data Types

The general properties of a data type are grouped on the data type's **Settings** tab. The following is the list of these properties:

Global Data Type: General Properties	
Property	**Description**
Title	A user-friendly name of a global data type used in most GUIs in C1. This name appears in the Data Navigator when the data type is saved.
Type name	A technical name of a global data type mostly used in code. It cannot be changed once used externally.
Type namespace	A namespace to group several data types together. This name appears in the Data Navigator if new when the data type is saved. Otherwise, the data type appears under the existing namespace specified.
Has publishing	When checked, this option indicates that the data items added to this data type will need publishing manually. When cleared, it indicates that the data items will be published automatically once saved.
Is localizable data	This option appears only if two or more locales (languages) are installed in C1. When checked, this option indicates that the data type can be localized.

2.4   Making Global Data Types Available in Content Perspective

Editors do not need to necessarily have access to the Data perspective in C1, particularly if they do not create data types but only use them. For this purpose, the existing global data types can be made available in the Content perspective.

To make global data types available in the Content perspective:

1.   In the Data Navigator, locate and right-click the data type you need.

2.   In the context menu, click **Show in Content perspective**.
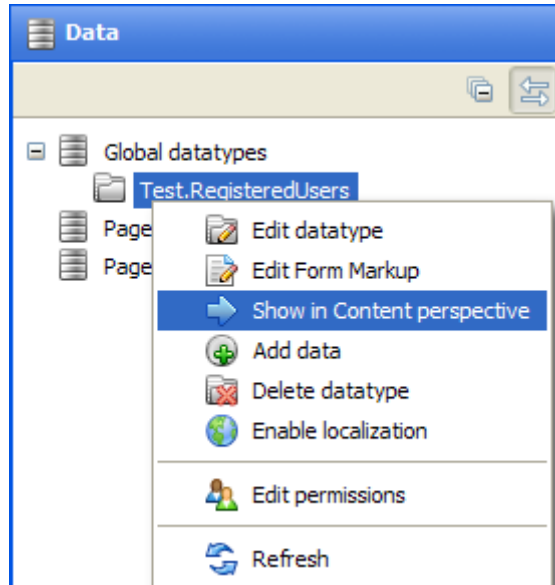
compo■ite

Figure 6: Show in Content perspective

3. Switch to the **Content** perspective. The global data type is now available here under **Data**.
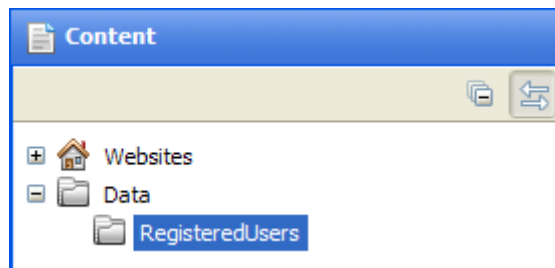

Figure 7: Data Type in Content perspective

If you right-click the data type in the **Data** perspective again, you will see a check mark next to **Show in Content perspective**, which indicates that this data type is already available in the **Content** perspective.
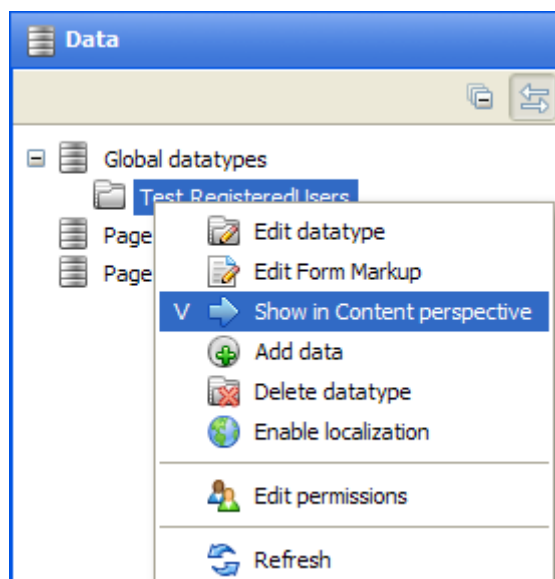

Figure 8: Indication of the data type available in Content perspective

composite

By clicking this menu command again, you will remove this data type from the Content perspective. The data type itself will remain untouched in the Data perspective.

2.5   Adding Data Items

Once you have created a global data type, you should fill it with structured data that will be available across your websites on all the web pages.

You can fill the global data type by adding data items to it. When adding a data item, you enter values in the fields added to this data type. Each data type has its own set of fields.

You can add data items not only in the Data perspective but also in the Content perspective if you make the data type available in the Content perspective.

By using input forms on the web pages, you can allow the users who visit your websites to fill the data types with data from their web browsers.

To add a data item:

1. In the Data Navigator, locate and select the data type you want to add a data item to.

2. Click **Add Data Item** on the toolbar. The data type item editor opens in the working area.

3. Fill out the data type fields.

Figure 9: Adding a data item

4. Click **Save**. The data item will appear under this data type.
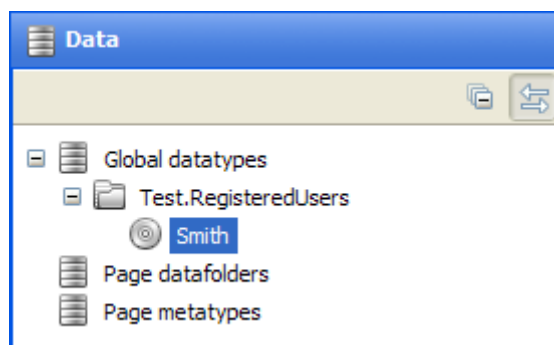

Figure 10: A data item added

A Guide to Creating Data Types

3　Creating and Managing Fields

Each data type might have one or more fields. You should add at least one field when creating a data type. The number and types of the fields you can add depend on your purposes.

For example, if you have a form on a web page where the user enters his personal data, you might have text fields for her or his first and last names, email address and phone number, a large text field (text area) for her or his address, and a date field for her or his date of birth.

When you add a field to a data type, you work with its settings editor in the working area. Each field has its basic and advanced properties grouped each on its corresponding tab in the settings editor.

You should set most of the basic properties manually while the advanced properties are set automatically for you with default values. However, if you want to fine-tune a field, you will probably set the advanced properties manually, too. For example, you can select another widget for the field.
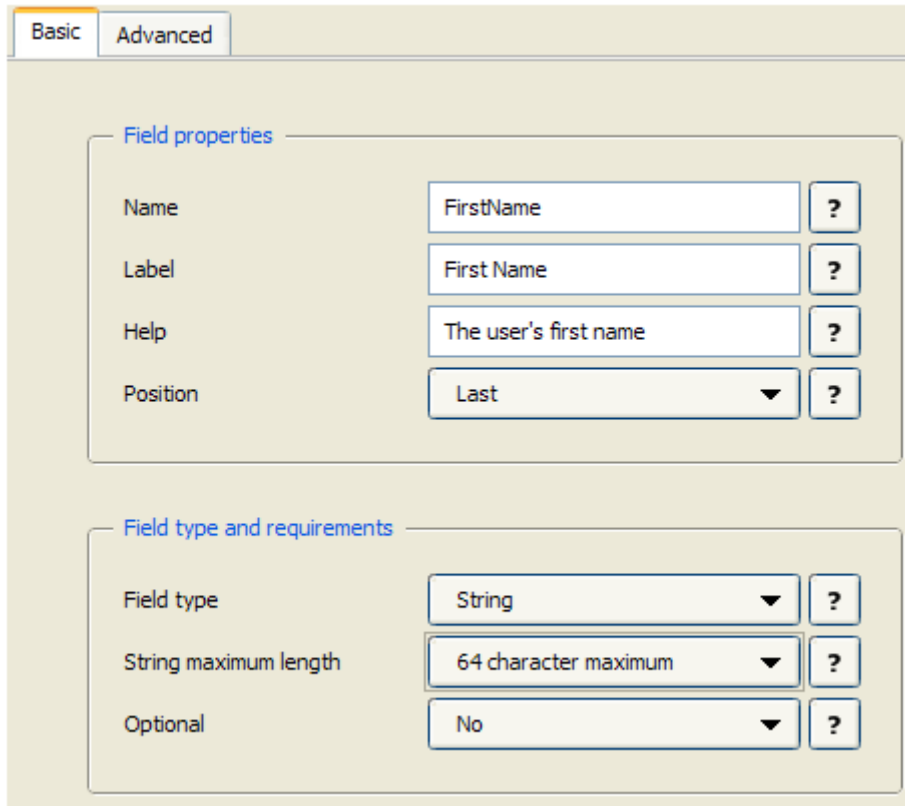
Each field you add should be of a certain type. The type determines what kind of data the field can hold. Based on the field type, some of the field's properties may differ. For example, for a string type you should specify the string's maximum length, for a decimal type – the number of decimal places, and for a data reference – its reference type.

The field type also determines what kind of a widget will be used for this field on the form. For example, with a string field intended for short texts, a text box widget will be most likely used, while the date field will use a date selector widget.

Let's first find out what basic and advanced properties you can set when adding a field to a data type with a closer look at field types.

compo**:**ite

3.1 Basic Field Properties

When you add a field, you should set a number of its basic properties such as its name, label, help text, position and type as well as indicate whether this field is required or optional.

Figure 11: Basic Field Properties

The following is the list of these basic properties:

Field Types	
Property	**Definition**
Name	The name of the field used by C1 to identify this field. The field cannot contain the space character or any other special characters.
Label	The name of the field displayed to the user in GUIs as a field label. It appears as a field label when adding a data item to a data type or on the forms based on these data types. If you do not specify the label it will copy the value of the Name field.
Help	Short help text associated with this field. It pops up when the user clicks the help button next to the field on a form.
Position	The position of the field with regards to other fields. By default, the Position is set to Last, which means the field will be added in the end of the field list.
Field Type	The type of the field that determines what data this field can hold.
Optional	This property indicates whether the input in this field is required or optional.

compo**:**ite

A Guide to Creating Data Types

Based on the field type, you might need to set an additional property specific to this type (e.g. *String maximum length*). These additional properties are discussed in the following section dedicated to the field types.
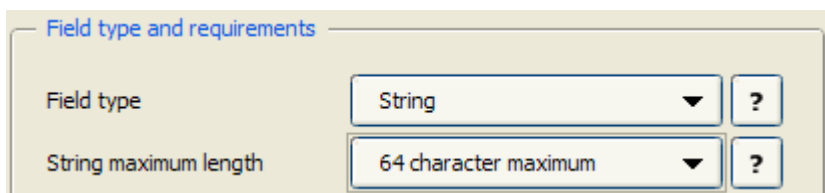
3.2   Field Types

For each global data type, C1 allows you to create fields of the following types:

- String
- XHTML
- Integer
- Decimal number
- Date
- Boolean
- Unique Identifier (GUID)
- Data reference

When you select the string, decimal number or data reference for the field type, you might also set an additional property specific to this type.

Normally, C1 sets this property to a default value but you can select a different value for it, too.

For the **String** type, you can also specify the string's maximum length. You can select *16*, *32*, *64*, *128*, *256*, *512* and *1024* characters for the maximum or set the string to the *unlimited length*. By default, the value is set to *64*.
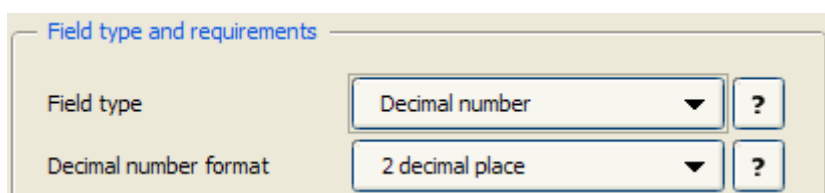

Figure 12: Additional property for the String field

For the **Decimal number** type, you can also set up the number of decimal places. You can choose *1*, *2*, *3* or *4*. The default value is set to *2*.
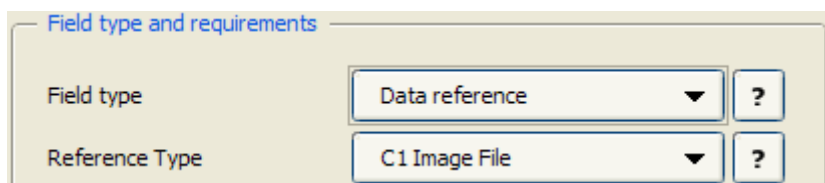

Figure 13: Additional property for the Decimal number field

For the **Data references**, you should select the reference type, that is, what entity you want to reference in this field. This will result in a special field that will serve as a list of these entities.

Figure 14: Additional property for the Data reference field

compo⬛ite

The data references can be of the following types:

Data Reference Types	
Property	**Description**
C1 Image File	Image files available in the C1 Media Archive
C1 Media File	Media files available in the C1 Media Archive
C1 Media Folder	Folders available in the C1 Media Archive
C1 Page	Pages created for one or more websites in C1

By default, the reference type is set to *C1 Image*.

In addition to these predefined reference types, if you have global data types, you can also select them for the reference type. In this case, the data items of this data type will be listed as entities in the field of this type.

3.3 Advanced Field Properties

A number of advanced properties allow you to fine-tune the field you add to a data type.

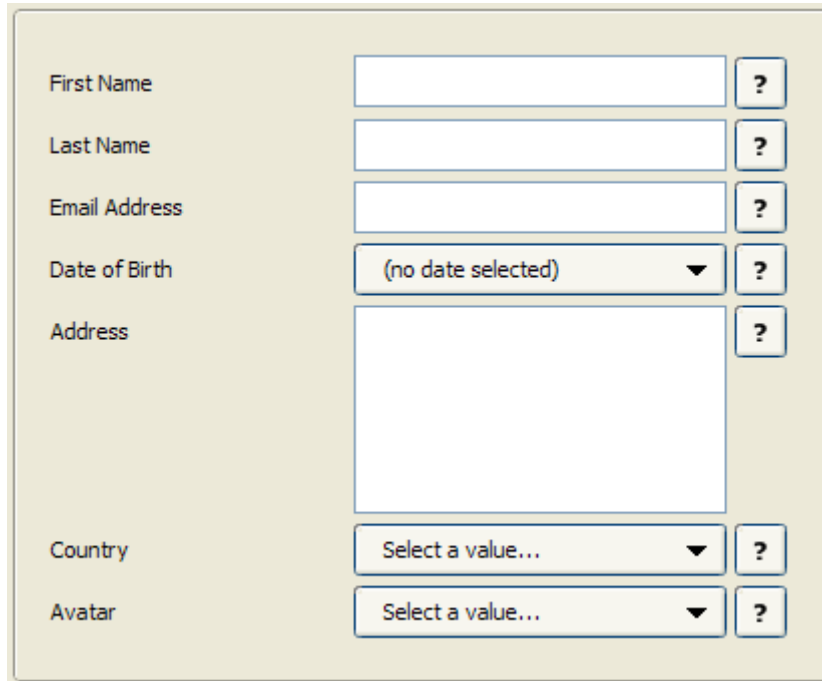Figure 15: Advanced field properties

Every time you add another field, its advanced properties are set to defaults automatically.
You can choose to override these default values.

Field Types	
Property	**Description**
Widget type	A widget to serve as an input control on a form representing this data type.
Validation rules	Constraints and limitations you can apply to the data entered in this field.
Is title field	When checked, it indicates that this field will be used as the title field when listing data in views like tree structures etc.
Tree ordering	When the *Order ascending (A-Z)* or *Order descending (Z-A)* option is selected, this field will be used to sort data in lists in the order specified. Based on the number of fields in the data type, you can also select this field to *Order 2 ascending* or *Order 2 decending* and so on.
Tree grouping	When the *Group by this field* option is selected, this field will be used to group data in long lists, which improves their readability. Based on the number of fields in the data type, you can also select this field to *Group as 2nd priority* and so on.
Default value	The value, to which the field will be set by default when a new data item is added.

compo*ite

A Guide to Creating Data Types

4   Widgets

Widgets are controls used in input forms that serve as form fields. You can see widgets when you add data items in C1 or in input forms on web pages. They get values the user enters and pass these values to the data type fields they are mapped to. Each widget has a specific field behind it. The type of the widget depends on the type of this field.

Figure 16: Widgets

When you add fields to a data type, C1 sets a default widget for this field. However, you can replace the default widget with some other widget available for this type of the field.

Some simple field types might only have one widget associated with it (for example, the integer field), while other field types can have a set of widgets you can choose from (for example, the string field).

Let's start with learning how to set a widget for a field and go on to have a look at the widgets available in C1.

4.1   Quick Search in Dropdown Lists

The selector widgets with dropdown lists allows you to quickly find an item in the list.

Start typing the word you want to find in the list, and the list will drop down, filter out all non-matching items and pre-select the first matching item.
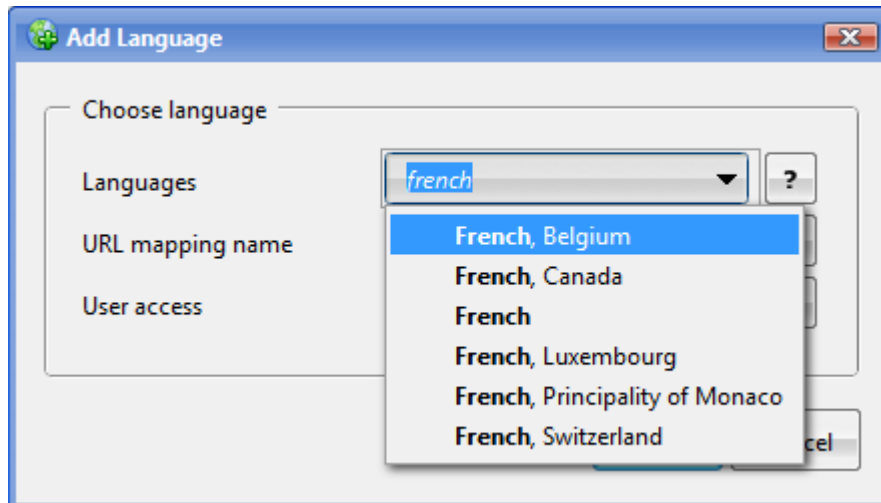
Figure 17: Quick searching in dropdown lists

4.2   Selecting Widget for Field

When you add a field and set its basic and advanced properties, you might also need to select a proper widget to render this field. C1 selects a default widget for the field you add. You can however replace the default widget with some other widget if more are available for this type of a field.

To select a widget for a field:

1. Select a field you want to edit.

2. On the **Advanced** tab, click the **Widget type** field to open the **Field Widget Configuration** window. It already has a widget selected.
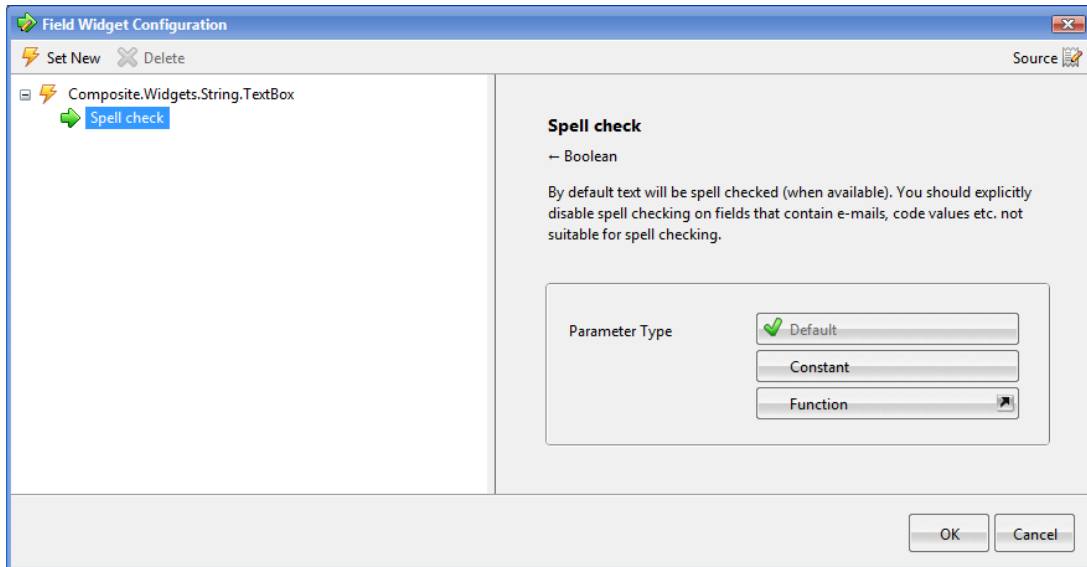
Figure 18: Currently selected widget

3. Click **Set New** to open the **Select Function** window. Here, you can see all the widgets available for this specific field.
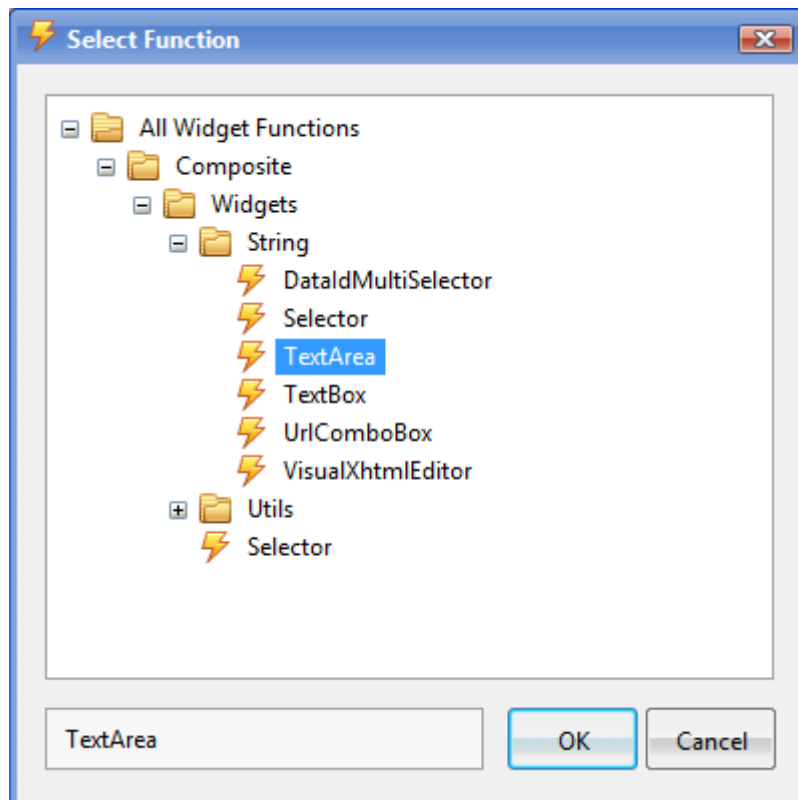
Figure 19: Selecting a different widget

A Guide to Creating Data Types

4. Locate and select the widget you want to set for the field and click **OK**.

5. Click **OK** in the **Field Widget Configuration** window.

The **Widget type** field is now set to the widget you have selected.
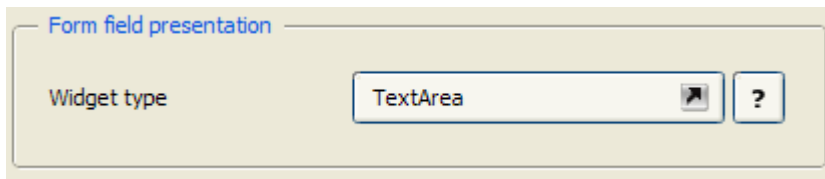
Figure 20: Newly selected widget

4.3   Overview of C1 Widgets

C1 features a number of widgets that can be used with one or more data type fields:

- TextBox
- TextArea
- DateSelector
- DateTimeSelector
- CheckBox
- BoolSelector
- VisualXhtmlEditor
- DataIdMultiSelector
- Selector
- Optional Selector
- ImageSelector
- MediaFileSelector
- MediaFileFolderSelector

Let's have a closer look at some of these widgets.

4.3.1   TextBox

The TextBox widget serves as an input control for texts strings and numbers.

Figure 21: TextBox

This widget is available for the String, Integer, decimal and GUID fields:

- Composite.Widgets.String.TextBox
- Composite.Widgets.Integer.TextBox
- Composite.Widgets.Decimal.TextBox
- Composite.Widgets.Guid.TextBox

The TextBox widget is default for the String, Integer, Decimal and GUID fields.

4.3.2   TextArea

The TextArea widget serves as an input control for large texts.

Figure 22: TextArea

This widget is only available for the String field:

- Composite.Widgets.String.TextArea

4.3.3 UrlComboBox

The UrlComboBox allows you to specify absolute URIs as well as select a C1 page or C1 media file via the Select Page or File dialog box.
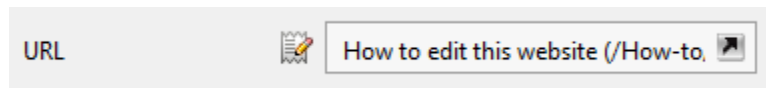

Figure 23: UrlComboBox

To edit the value as a text string, you need to click the Edit button on the left of the widget. To select a page or file, you should click the button in the widget on the right.

This widget is only available for the String field:

- Composite.Widgets.String.UrlComboBox

4.3.4 DateSelector

The DateSelector widget allows the user to select a date as an input value. As the user clicks the widget, a month calendar drops down where the user can navigate to the specific year and a month and select a date.
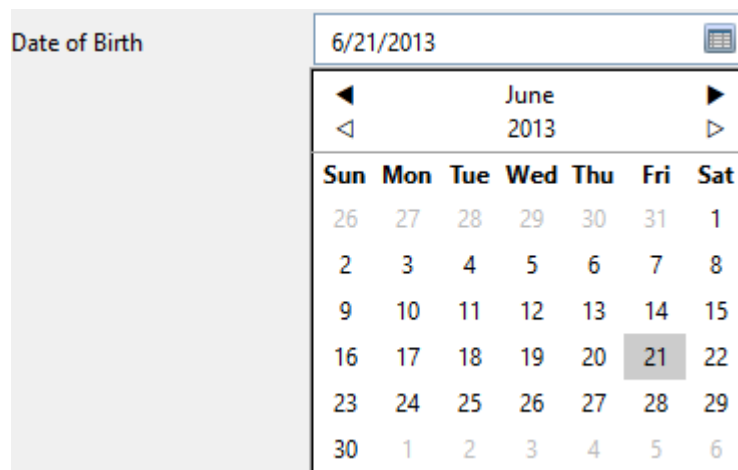
Figure 24: DateSelector

This widget is only available for the Date field:

- Composite.Widgets.Date.DateSelector

The DateSelector widget is default for the Date field.

A Guide to Creating Data Types

4.3.5   DateTimeSelector

The DateTimeSelector widget allows the user to select not only a date but also time as an input value. As the user clicks the widget, a month calendar drops down where the user can navigate to the specific year and a month and select a date. When the date is set, the user can edit the time in the widget.
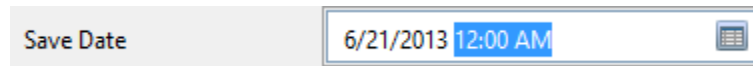

Figure 25: DateTimeSelector

This widget is only available for the Date field:

- Composite.Widgets.Date.DateTimeSelector

4.3.6   CheckBox

The CheckBox widget serves as an input control for selecting one of the two opposite options. Switching between the two options implies checking and unchecking the check box.
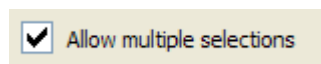
Figure 26: CheckBox

This widget is only available for the Boolean field:

- Composite.Widgets.Bool.CheckBox

The CheckBox widget is default for the Boolean field.

4.3.7   BoolSelector

The BoolSelector widget is a variation of the CheckBox widget in its purpose. While the checkbox allows the user to switch between two options by checking and unchecking the check box, the BoolSelector widget allows the user to select one of the two radio buttons representing two options.
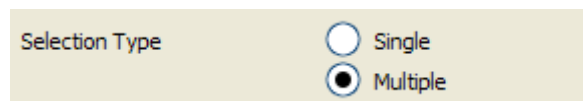
Figure 27: BoolSelector

This widget is only available for the Boolean field:

- Composite.Widgets.Bool.BoolSelector

4.3.8   VisualXhtmlEditor

The VisualXhtmlEditor widget allows the user to launch Visual Editor to specify XHTML-formatted text as an input value for a string field.
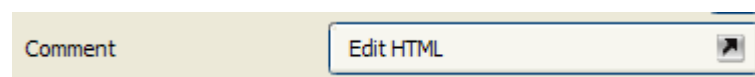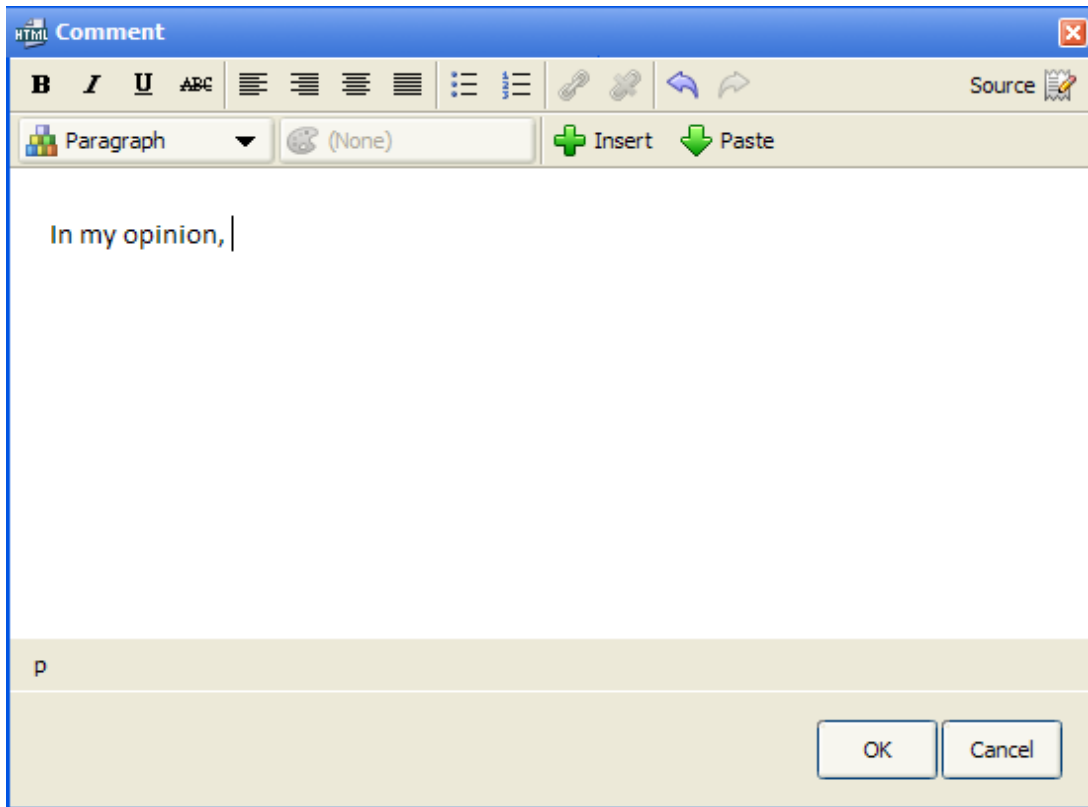

Figure 28: VisualXhtmlEditor

Figure 29: Launched Visual XHTML Editor

This widget is only available for the string field:

- Composite.Widgets.String.VisualXhtmlEditor

4.3.9 DataIdMultiSelector

The DataIdMultiSelector widget serves to present a list of input values available for multiple choices. The input values represent entities such as C1 image files, C1 media files, C1 media folders, C1 pages or data items of available data types, which is configured in the widget's parameters.

It comes in two UI views: *verbose* and *compact*, which is also configurable in the widget's parameters.

The **verbose UI view** lists values with check boxes next to them. In this way, the user can select one or more values.
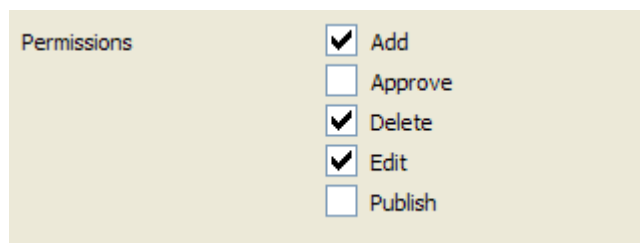
Figure 30: DataIdMultiSelector: Verbose UI View

The **compact UI view** allows the user to launch the selection editor where the user can select one or more values moving them from the list of available values to the list of selected ones.
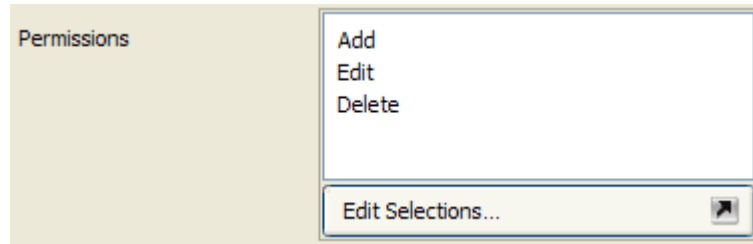
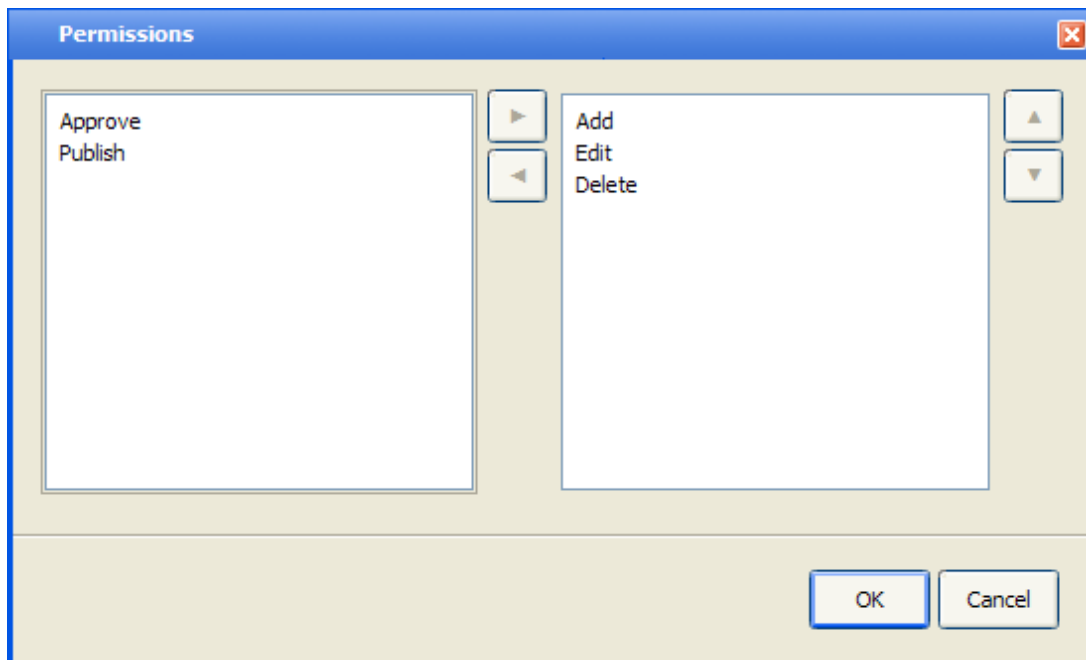Figure 31: DataIdMultiSelector: Compact UI View

Figure 32: DataIdMultiSelector: Selecting values

This widget is only available for the string field:

- Composite.Widgets.String.DataIdMultiSelector

4.3.10   Selector

The Selector widget serves to present a list of input values. For most field types it is associated with, it shows a drop down list where the user can only select one value.
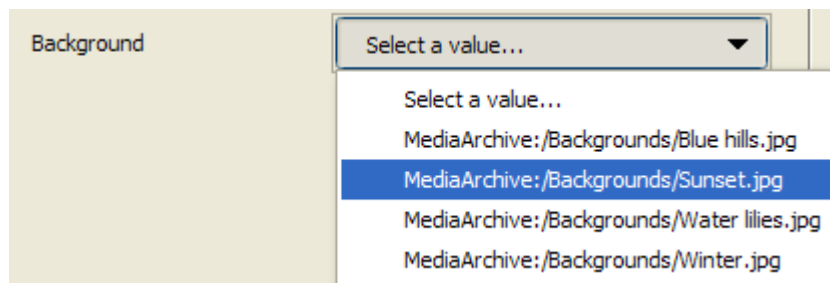

Figure 33: Selector

This widget is available for the string field and data reference fields of different reference types (including available data types):

- Composite.Widgets.String.Selector
- Composite.Widgets.CompositeDataTypesIImageFile.Selector

- Composite.Widgets.CompositeDataTypesIMediaFile.Selector
- Composite.Widgets.CompositeDataTypesIMediaFileFolder.Selector
- Composite.Widgets.CompositeDataTypesIPage.Selector

The Selector widget is default for data type reference types.

4.3.11   OptionalSelector

Like the Selector widget, the OptionalSelector widget serves to present a list of input values. However, it includes another value: *<NONE>*, which allows the user not to select any of the values available.
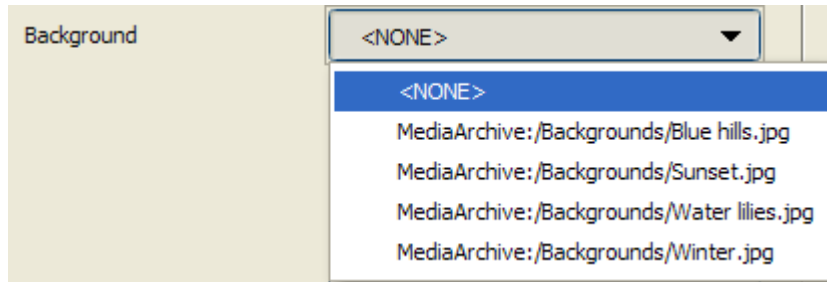

Figure 34: OptionalSelector

This widget is available for the string and data reference fields of different reference types (including available data types):

- Composite.Widgets.String.OptionalSelector
- Composite.Widgets.CompositeDataTypesIImageFile.OptionalSelector
- Composite.Widgets.CompositeDataTypesIMediaFile.OptionalSelector
- Composite.Widgets.CompositeDataTypesIMediaFileFolder.OptionalSelector
- Composite.Widgets.CompositeDataTypesIPage.OptionalSelector

4.3.12   Other Selector-Like Widgets

Apart from the Selector and OptionalSelector widgets you can use with the Data Reference fields, there are specific selector-like widgets available for C1 Page, C1 Image File, C1 Media File and C1 Media Folder reference types:

- PageSelector
- ImageSelector
- MediaFileSelector
- MediaFileFolderSelector

Each of these widgets is default for the Data Reference field of the corresponding reference type.
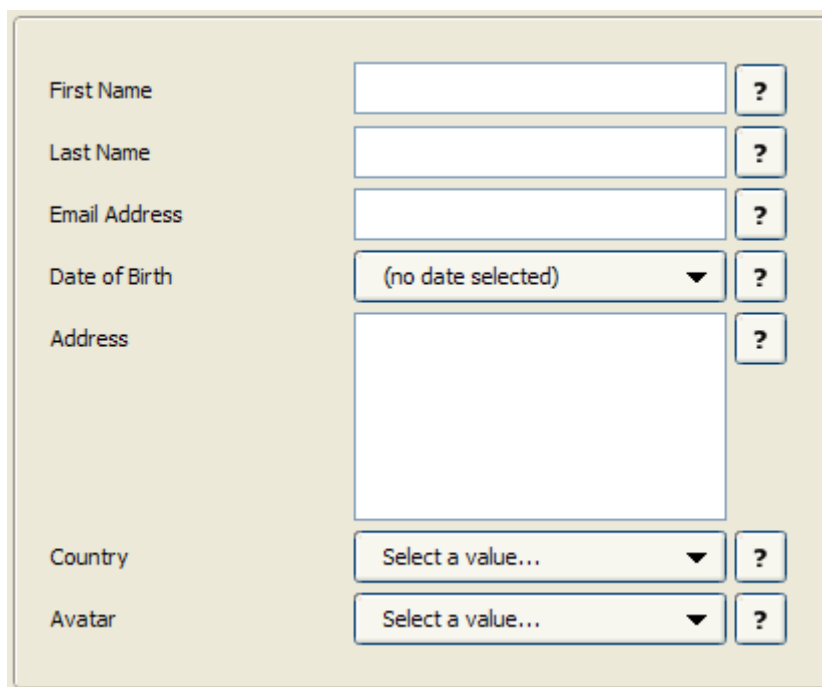
5   Creating Specific Fields

Now that you have learned how to create data types and add fields to these data types, set properties and select widgets for the fields, let's see how to create fields that will serve specific purposes.

Let's assume that you have a form that requires users to fill the following data:

- First name
- Last name
- Email address
- Address
- Date of birth
- Avatar
- Country

For this purpose you need to have a data type with a number of specific fields.

First of all, you will need string fields for the first 4 items. However, one of them - the Address field - must be large enough to fit an address that might be quite long. Then, you will use the Date field for the user's date of birth. For the Avatar, you will allow the user to select one of the images available to serve as her or his graphic representation. Finally, you will create a very simple list of several countries the user will be able to choose from.

Figure 35: Form to create

With this purpose in mind, let's see how to create:

- A simple text field
- A large text field
- A Date field
- A Data Reference field
- A String field with the Selector

5.1   How to Create Simple Text Field

To create a simple text field:

1. Add a field to the data type.

2. Set its basic properties such as its *name*, *label*, *help text* and *position*.

3. For the **Field type**, select *String* and specify the reasonable *length maximum*, e.g. *32*.

4. Specify whether this field is optional or required.

Figure 36: Simple Text Field: basic properties

5. On the **Advanced** tab, make sure the *TextBox* is selected as the **Widget type**.

A Guide to Creating Data Types

6. If necessary, specify other advanced properties.

Figure 37: Simple Text Field: advanced properties

7. Click **Save**.

Now this field will be available on an input form as a simple text box.

5.2   How to Create Large Text Field

To create a large text input field (also known as a text area):

1. Add a field to the data type.

2. Set its basic properties such as its *name*, *label*, *help text* and *position*.

3. For the **Field type**, select *String* and specify the reasonable *length maximum*, e.g. *unlimited length*.

composite

4. Specify whether this field is optional or required.

Figure 38: Large Text Field: basic properties

6. On the **Advanced** tab, click the **Widget type** field to open the **Field Widget Configuration** window.

Figure 39: Field Widget Configuration window

7. Click **Set New** to open the **Select Function** window and locate and select the **TextArea** widget (**Composite.Widgets.String.TextArea**).
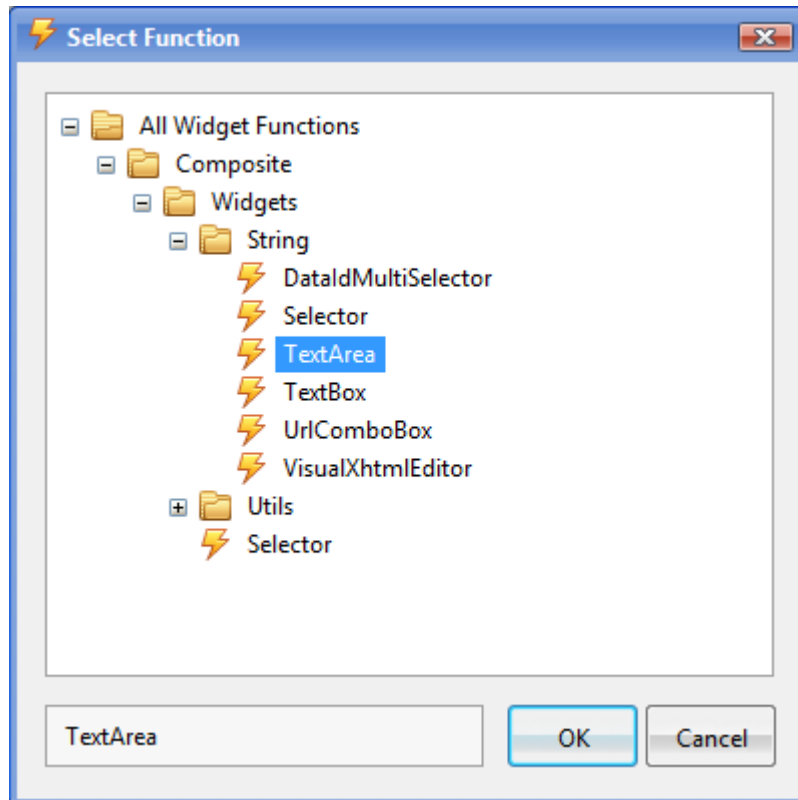
A Guide to Creating Data Types

compo=ite

Figure 40: Selecting TextArea widget

8. Click **OK** in the **Select Function** window and in the **Field Widget Configuration** window.

9. If necessary, specify other advanced properties.

Figure 41: Large Text Field: advanced properties

10. Click **Save**.

Now this field will be available on an input form as a text area.

5.3   How to Create Date Field

To create a date field that will allow the user to select a date for its value:

1. Add a field to the data type.

2. Specify its basic properties such as its *name*, *label*, *help text* and *position*.

3. For the **Field type**, select *Date*.

4. Specify whether this field is optional or required.

Figure 42: Date Field: basic properties

5. On the **Advanced** tab, make sure that the *DateSelector* is selected as the **Widget type**.

A Guide to Creating Data Types

compo**s**ite

6. If necessary, specify other advanced properties.

Figure 43: Date Field: advanced properties

7. Click **Save**.

Now this field will be available on an input form as a date selector.

5.4   How to Create Data Reference Field

For the Data Reference field we will use the C1 Image File reference type.

To create a Data Reference field of the C1 Image File type:

1. Add a field to the data type.

2. Specify its basic properties such as its *name*, *label*, *help text* and *position*.

3. For the **Field type**, select *Date Reference* and for the **Reference Type** – *C1 Image File*.

A Guide to Creating Data Types

4. Specify whether this field is optional or required.

Figure 44: Data Reference Field of C1 Image type: basic properties

5. On the **Advanced** tab, make sure that the *ImageSelector* is selected as the **Widget type**. (Please note that you cannot use the ImageSelector widgets with Composite.Forms.Renderer, in which case you should replace them with the corresponding Selector or Optional Selector widget. Please see Using Input Forms with Data Types for more information.)

compo⬚ite

6. If necessary, specify other advanced properties.

Figure 45: Data Reference Field of C1 Image type: advanced properties

7. Click **Save**.

Now this field will be available on an input form as a drop down list filled with titles of the images available in the Media Archive.

In the similar manner, you can create data reference fields for web pages, media files (e.g. Flash videos), media folders and data items available in other data types.

5.5   How to Create String Field with Selector

In some cases, you do not need to create a separate data type, the data items of which will be used in the selector widget on the form. All you have to do is to provide the list of options separated with a comma (or another predefined separator) in a single string and use it with a simple Selector widget for the String field.

To create a string field that will allow the user to select options from a drop down list:

1. Add a field to the data type.

2. Specify its basic properties such as its *name*, *label*, *help text* and *position*.

3. For the **Field type**, select *String* and specify the reasonable *length maximum*, e.g. *Unlimited length*.

4. Specify whether this field is optional or required.

Figure 46: The String field: basic properties

5. On the **Advanced** tab, click the **Widget type** field to open the **Field Widget Configuration** window.

6. Click **Set New** to open the **Select Function** window and locate and select the *Selector* widget. Click **OK**.
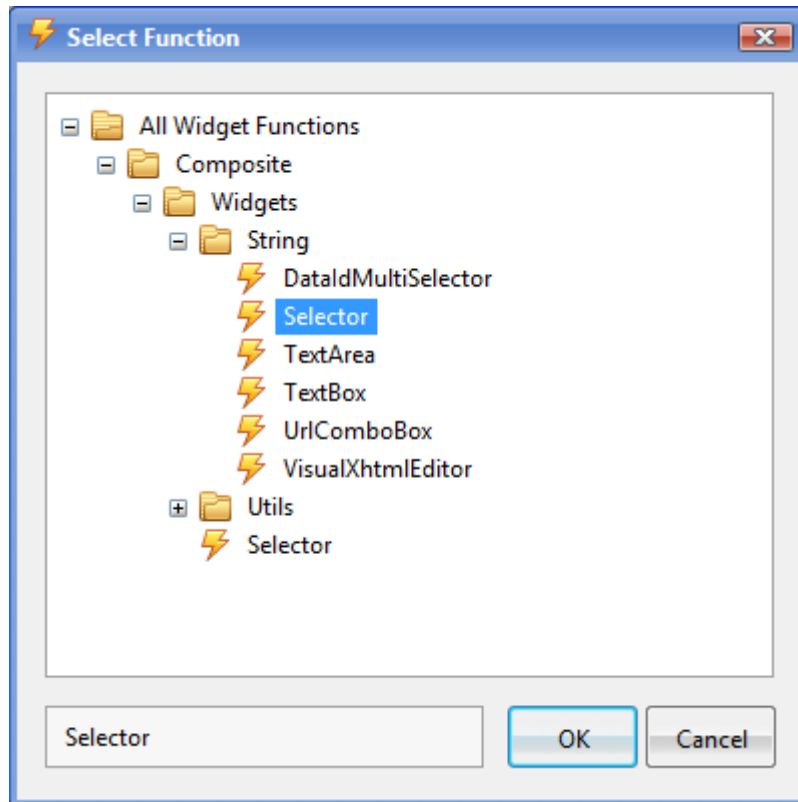
Figure 47: Selecting the Selector widget

7. Select the **Options** parameter of the **Selector** widget and click *Function* for its **Parameter type**.
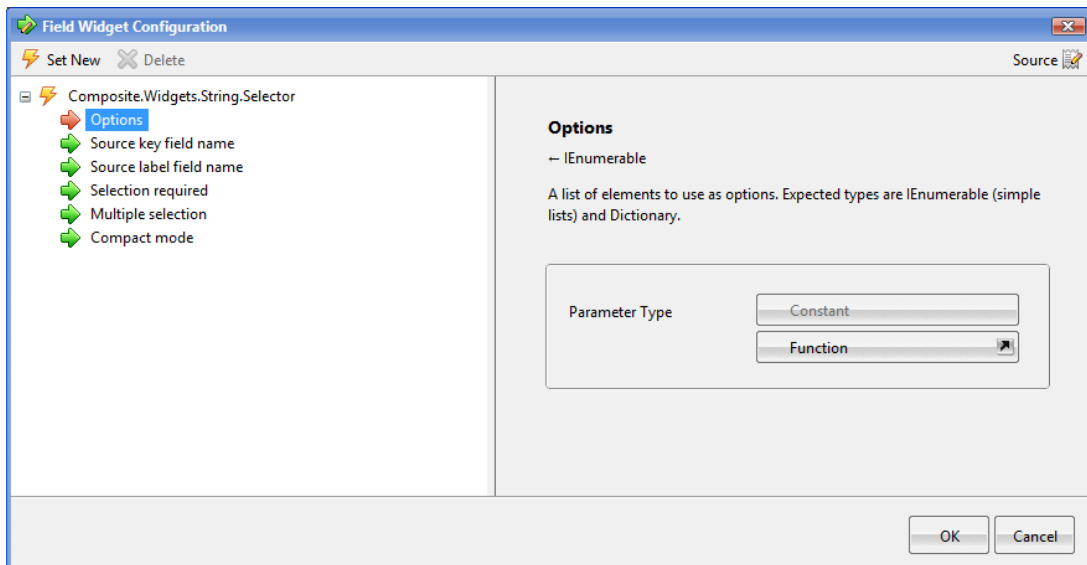
Figure 48: Setting the Options parameter

8. Locate and select the **Composite.Utils.String.Split** function in the **Select Function** window. Click **OK**.
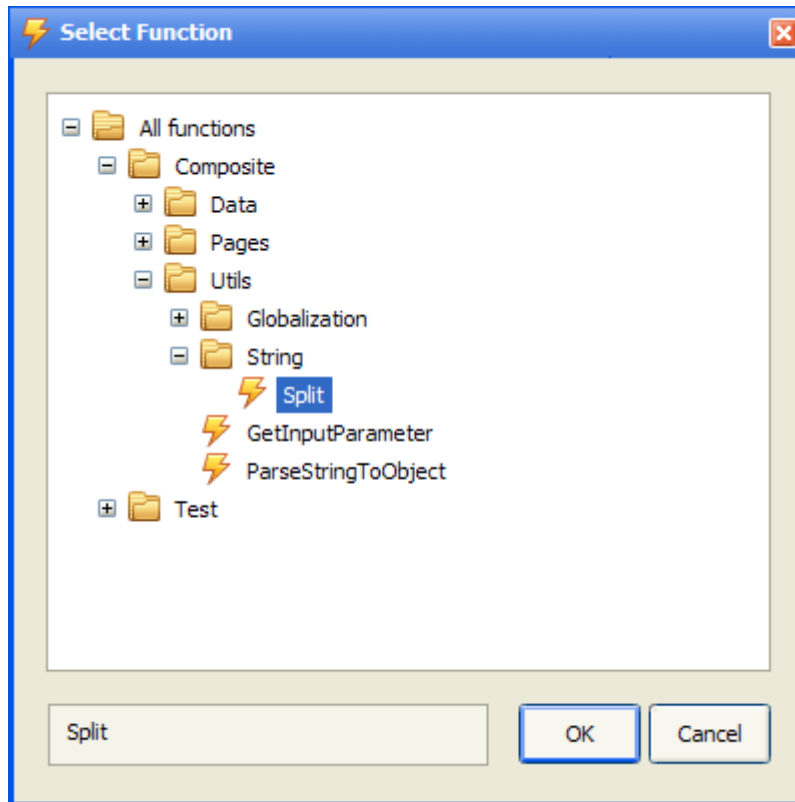
Figure 49: Selecting the **Composite.Utils.String.Split** function

9. In the **String to split** parameter of this function, enter a few options separated by a comma, e.g. "*Denmark,USA,UK,The Netherlands,Germany*". Click **OK**.
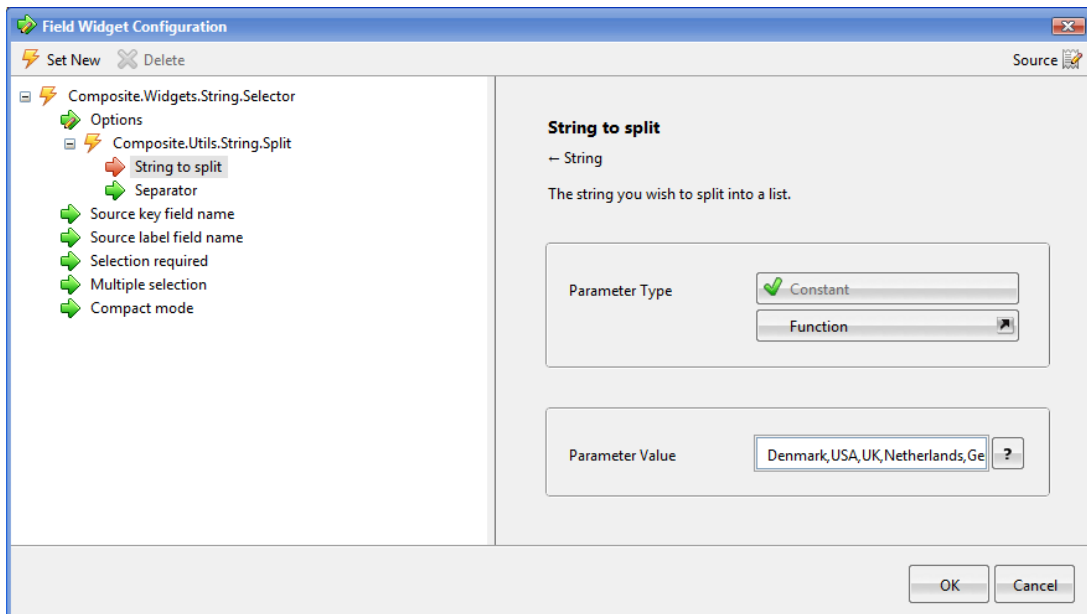
Figure 50: Specifying a string to split

10. Click **OK** in the **Field Widget Configuration** window.

A Guide to Creating Data Types

11. If necessary, specify other advanced properties.

Figure 51: The String field: advanced properties

12. Click **Save**.

Now this field will be available on an input form as a drop down list filled with options you have specified.

6 Using Input Forms with Data Types

To allow the visitors of your website to add data to your data types from their web browsers, you should add input forms to web pages.

The **Composite.Forms.Renderer** package can create these input forms automatically and handle the input information for you. All you have to do is to add the Composite.Forms.Renderer function to a web page and link it to a global data type that will hold the input information.

The data added via these forms will be added to the corresponding data type and can be read or used from within the C1 application.

You can set up this Composite.Forms.Renderer function to receive email notifications every time a person submits data via the form. CAPTCHAs can help you avoid automatic form filling by spam bots etc.

6.1 Adding Input Form to Web Page

Before you add an input form to one of your web pages, you should:

1. Make sure that the **Composite.Forms.Renderer** package has been installed in your instance of Composite C1.

2. Have a *global data type* available to hold data entered by the user via the form with all necessary fields added and set up.

Please note that if your data type contains Data Reference fields of the C1 Page, C1 Image, C1 Media File, or C1 Media Folder reference type, you should change their default widgets to the corresponding Selectors or Optional Selectors. Currently Composite.Forms.Renderer does not support these widgets. Change the widgets as follows:

- C1 Image: Composite.Widgest.DataReference.DataTypesIImageFile.Selector
- C1 Page: Composite.Widgest.DataReference.DataTypesIPage.Selector
- C1 Media File: Composite.Widgest.DataReference.DataTypesIMediaFile.Selector
- C1 Media Folder: Composite.Widgest.DataReference.DataTypesIMediaFileFolder.Selector

To add an input form to a web page:

1. Edit a web page where you want add the form.

2. On the page's toolbar, click **Insert**, **Function** to open the **Select Function** window.

3. Locate and select the **Composite.Forms.Renderer** function and click **OK**.

4. In the **Function Properties** window, for the **DataType** property, select the data type that will hold the input data for this form. All the fields in this data type will become available in the form via the widgets you selected for this data type.
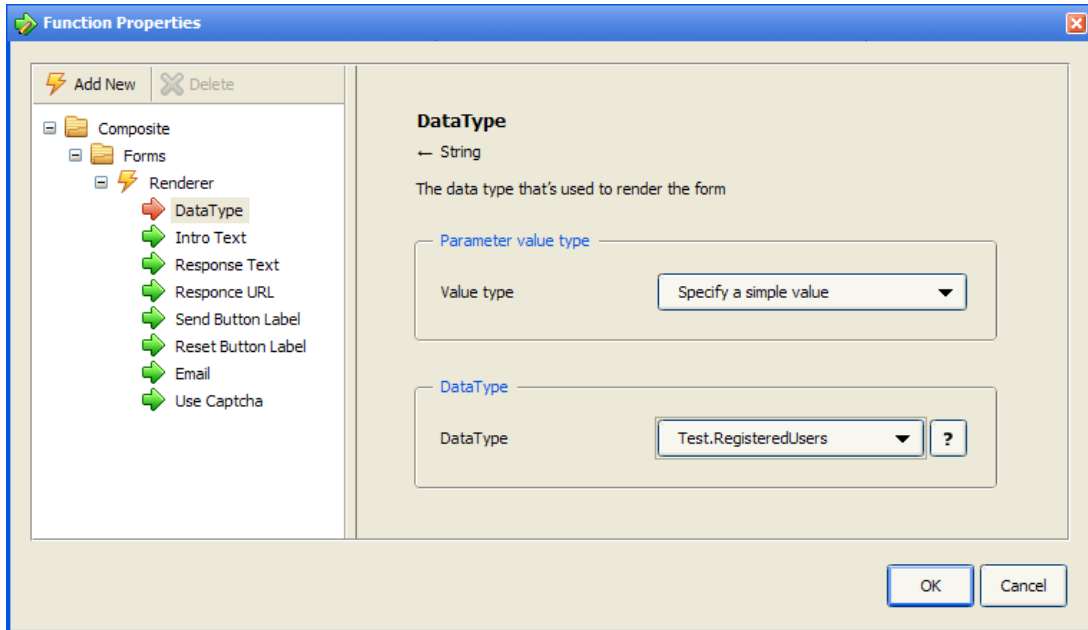
Figure 52: Setting up **Composite.Forms.Renderer**

5. Set other optional properties such as **Intro Text**, **Response Text**, **Response URL**, **Send Button Label**, **Reset Button Label**, **Email**, **Use Captcha**.

6. Click **OK**.

7. Save and publish the page.

Now users can open this page in their web browsers and fill out the form. All the data entered in this form will be stored in the corresponding data type.

Figure 53: The input form on a web page

The following are the properties you should or can set up for the input form you create:

Composite.Forms.Renderer Function Properties	
Property	**Description**
DataType	A data type to be used to render the form. It will hold values entered via the form. Required.
Intro Text	An XHTML-formatted text to be displayed above the form before the user submits the form.
Response Text	An XHTML-formatted text to be displayed instead of the form after the user submits the form.
Response URL	A page the user is redirected to after she or he submits the form.
Send Button Label	The caption on the Send (Submit) button to be used instead of the default caption.
Reset Button Label	The caption on the Reset button to be used instead of the default caption.
Email	A number of parameters that will enable email notifications on submitting the form.
Use Captcha	When checked, it indicates that the form will require to enter the CAPTCHA text to protect from spam bots etc.

6.2   Setting Up Email Notifications

Optionally, if you would like to receive notifications every time a user submits a form on your website, you can set up email notifications for the **Composite.Forms.Renderer** used on a web page.

⚠ *Important: To be able to send email notifications, make sure that you have configured an email host by editing ~/web.config and including the /configuration/system.net/mailSettings section. For information about configuring the e-mail host in web.config, please refer to http://docs.composite.net/ConfiguringEmailHost .*

To send an email notification to one recipient:

1. Select the **Composite.Forms.Renderer** function on a web page and click to select its **Email** parameter.

2. Set its **Value type** to *Execute a function call* and the launch the function designer.

3. In the **Select Function** window, select the **Composite.Forms.FormEmailHeader** function and click **OK**.

4. Specify the parameters of this function and click **OK**:

    - **From**: The sender's email address.
    - **To**: The recipient's email address.
    - **Cc**: An optional field to specify other recipients' email addresses to send an email notification copy to.
    - **Subject**: The email message's subject

5. Click **OK** in the **Function Properties** window.

compo=ite

To send an email notification to two recipients:

1. Select the **Composite.Forms.Renderer** function on a web page and click to select its **Email** parameter.
2. Set its **Value type** to *Execute a function call* and the launch the function designer.
3. In the **Select Function** window, select the **Composite.Forms.JoinEmailHeader** function and click **OK**.
4. Now select the **Email A** parameter and repeat the above steps for setting up the **Composite.Forms.FormEmailHeader** function.
5. Select the **Email B** parameter and repeat the above steps for setting up the **Composite.Forms.FormEmailHeader function**, too.

compo**site**

A Guide to Creating Data Types

7 Test Your Knowledge

7.1 TASK 1

1. Create a global data type.
2. Specify its title and type name as "*RegisteredUsers*".
3. Enter "Test" for the type namespace.

7.2 TASK 2

1. Add 3 string fields to the RegisteredUsers data type and name them "*FirstName*", "*LastName*" and "*EmailAddress*" accordingly.
2. Add another string field and name it "*About myself*".
3. Set its Widget type to "*TextArea*".
4. Add a date field to the data type and name it "*DateOfBirth*".
5. Add a data reference field of the *C1 Image File* type and name it "*Avatar*".
6. Change its widget to *Composite.Widgest.DataReference.DataTypesIImageFile.Selector*

7.3 TASK 3

1. Add a string field and name it "*Country*".
2. Set its Widget type to "*Selector*".
3. Specify the Country options for the selector: "*Denmark*", "*USA*", "*UK*", "*The Netherlands*" and "*Germany*".
4. Save the data type.
5. Make the data type available in the Content perspective.

7.4 TASK 4

1. Create a web page and name it "*Registration*".
2. Add an input form to the web page (Composite.Forms.Renderer) and associate it with the *RegisteredUsers* data type.
3. Set its response text to "*Thank you for registering with us.*"
4. Make the form use the CAPTCHA.
5. Save and publish the page.

7.5 TASK 5

1. Add a data item to the *RegisteredUsers* global data type from the Content perspective.
2. Open the *Registration* page in a web browser.
3. Fill out and submit the form.

compo:ite

A Guide to Creating Data Types