



# A Guide to Creating Datatypes

Composite 2011-07-25

# Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	Who Should Read This Guide?	3
1.2	Getting Started	4
1.3	Terms and Abbreviations	4
<b>2</b>	<b>CREATING AND MANAGING DATATYPES .....</b>	<b>5</b>
2.1	Datatypes	5
2.2	Creating Global Datatypes	6
2.3	General Properties of Datatypes	9
2.4	Making Global Datatypes Available in Content Perspective	9
2.5	Adding Data Items	10
<b>3</b>	<b>CREATING AND MANAGING FIELDS .....</b>	<b>12</b>
3.1	Basic Field Properties	13
3.2	Field Types	14
3.3	Advanced Field Properties	15
<b>4</b>	<b>WIDGETS .....</b>	<b>17</b>
4.1	Selecting Widget for Field	17
4.2	Overview of C1 Widgets	19
4.2.1	<i>TextBox</i>	20
4.2.2	<i>TextArea</i>	21
4.2.3	<i>DateSelector</i>	21
4.2.4	<i>DateTimeSelector</i>	22
4.2.5	<i>CheckBox</i>	22
4.2.6	<i>BoolSelector</i>	23
4.2.7	<i>VisualXhtmlEditor</i>	23
4.2.8	<i>DataIdMultiSelector</i>	24
4.2.9	<i>Selector</i>	25
4.2.10	<i>OptionalSelector</i>	26
4.2.11	<i>Other Selector-Like Widgets</i>	26
<b>5</b>	<b>CREATING SPECIFIC FIELDS.....</b>	<b>27</b>
5.1	How to Create Simple Text Field	28
5.2	How to Create Large Text Field	29
5.3	How to Create Date Field	32
5.4	How to Create Data Reference Field	34
5.5	How to Create String Field with Selector	36
<b>6</b>	<b>USING INPUT FORMS WITH DATATYPES.....</b>	<b>41</b>
6.1	Adding Input Form to Web Page	41
6.2	Setting Up Email Notifications	43
<b>7</b>	<b>TEST YOUR KNOWLEDGE .....</b>	<b>45</b>

# 1 Introduction

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 datatypes in C1.

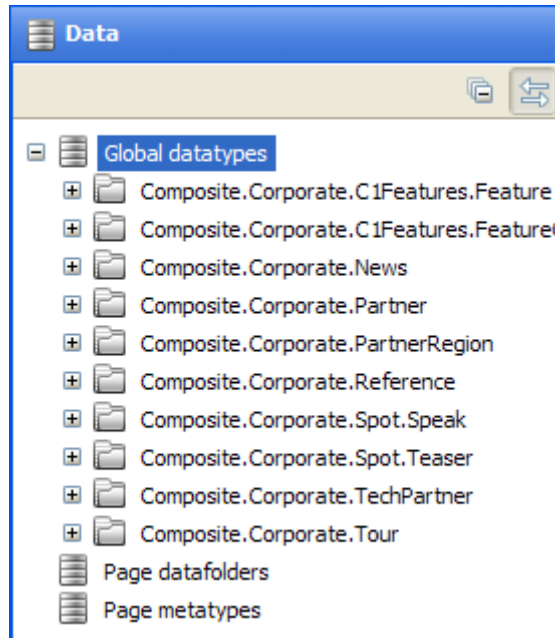


Figure 1: Datatypes in Data perspective

This guide focuses on creating datatypes that you can further use on web pages via input forms for entering data. It also gives an overview of datatype properties as well as datatype 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 datatypes 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 datatypes 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 datatype creator, you will primarily work in the Data perspective where you will need permissions to create and edit datatypes. As a datatype 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 datatypes, you are supposed to take a number of steps.

Getting Started		
Step	Activity	Chapter or section
1	Create global datatypes	<a href="#">2 Creating and Managing Datatypes</a>
2	Add fields to datatypes	<a href="#">3 Creating and Managing Fields</a>
3	Select widgets for datatype fields	<a href="#">4 Widgets</a>
4	Create specific fields for input forms	<a href="#">5 Creating Specific Fields</a>
5	Use input forms with datatypes	<a href="#">6 Using Input Forms with Datatypes</a>

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
Datatype	An entity used to store and reuse structured data in Composite C1
Global datatype	A datatype that represents data reusable across all web pages on all web sites created in C1.
Page datafolder	A datatype that represent data associated with one specific page.
Page metatype	A datatype 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 datatype fields

## 2 Creating and Managing Datatypes

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

### 2.1 Datatypes

C1 datatypes allows 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 datatypes:

- Global datatypes
- Page datafolders
- Page metatypes

The **Global datatypes** 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 datatype 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 datatypes*, which allow you to store and reuse data across multiple web pages.

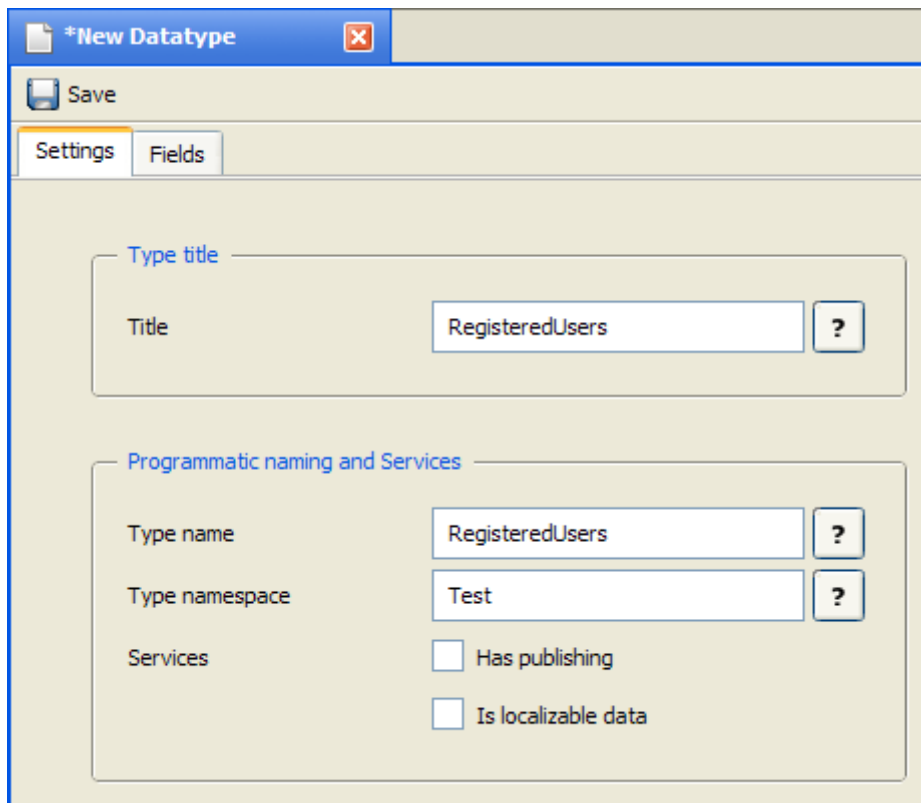
In the following few chapters and sections you will learn how to create global datatypes, how to set their properties, how to add fields to these datatypes and how to use forms with datatypes on web pages.

## 2.2 Creating Global Datatypes

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

### To create a global datatype:

1. In the Data Navigator, select **Global datatypes** and click **Add Datatype** on the toolbar. The datatype settings editor opens in the working area.
2. On the **Settings** tab, set the datatype's general properties such as its **Title**, **Type name**, **Type namespace**.



The screenshot shows a software interface for creating a new datatype. The window title is '\*New Datatype'. Below the title bar is a 'Save' button. There are two tabs: 'Settings' (selected) and 'Fields'. The 'Settings' tab contains two main sections:

- Type title**: A text input field labeled 'Title' containing the text 'RegisteredUsers' and a question mark icon.
- Programmatic naming and Services**:
  - A text input field labeled 'Type name' containing 'RegisteredUsers' and a question mark icon.
  - A text input field labeled 'Type namespace' containing 'Test' and a question mark icon.
  - Two checkboxes under the 'Services' section:
    - Has publishing
    - Is localizable data

Figure 2: General datatype 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.

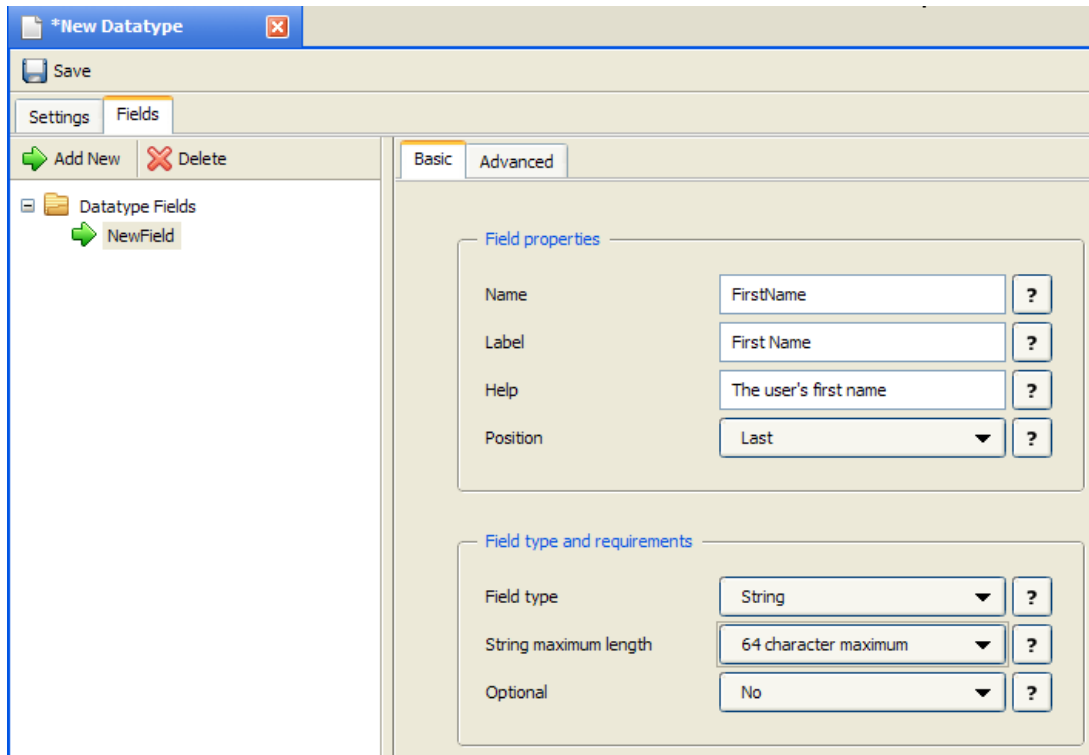


Figure 3: Basic field properties

4. If necessary, click the **Advanced** tab and set the [field's advanced properties](#).

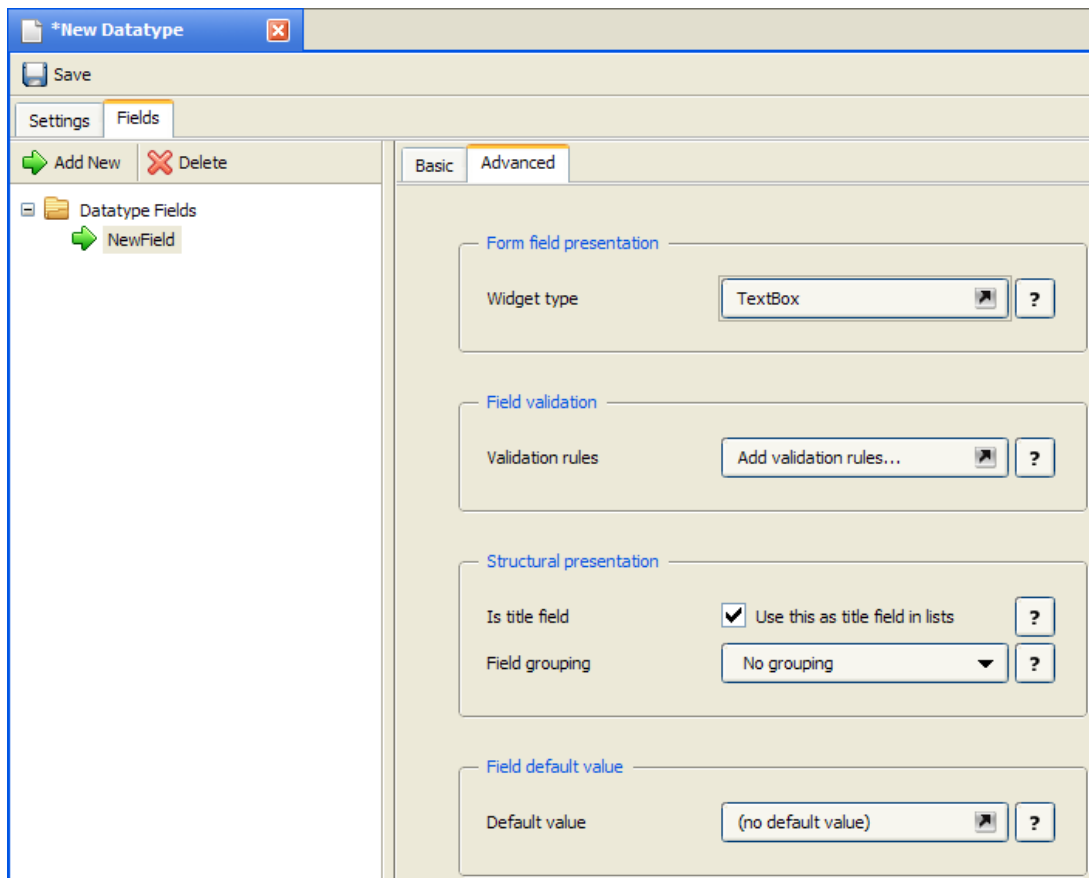


Figure 4: Advanced field properties

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

The datatype appears under Global datatypes in the Data Navigator.

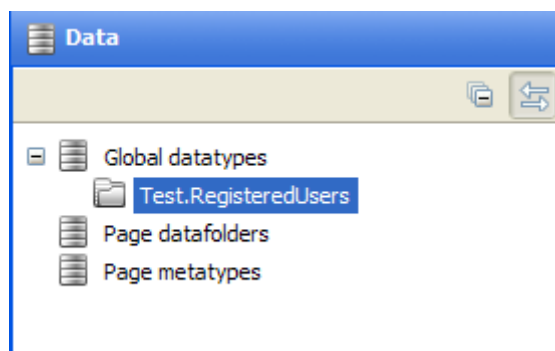


Figure 5: New global datatype



## 2.3 General Properties of Datatypes

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

Global Datatype: General Properties	
Property	Description
Title	A user-friendly name of a global datatype used in most GUIs in C1. This name appears in the Data Navigator when the datatype is saved.
Type name	A technical name of a global datatype mostly used in code. It cannot be changed once used externally.
Type namespace	A namespace to group several datatypes together. This name appears in the Data Navigator if new when the datatype is saved. Otherwise, the datatype appears under the existing namespace specified.
Has publishing	When checked, this option indicates that the data items added to this datatype 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 datatype can be localized.

## 2.4 Making Global Datatypes Available in Content Perspective

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

**To make global datatypes available in the Content perspective:**

1. In the Data Navigator, locate and right-click the datatype you need.
2. In the context menu, click **Show in Content perspective**.

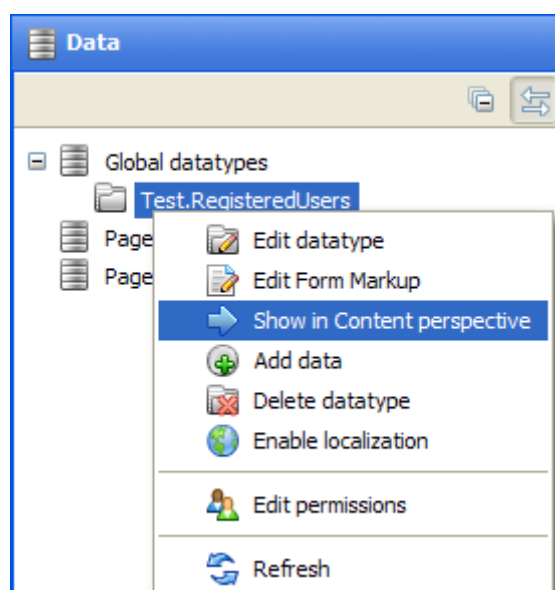


Figure 6: Show in Content perspective

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

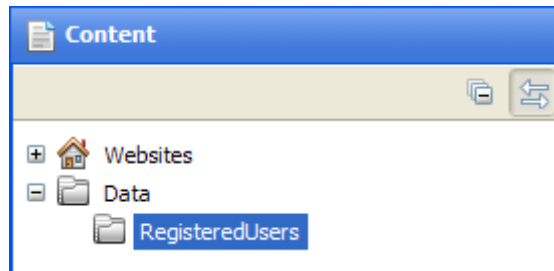


Figure 7: Datatype in Content perspective

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

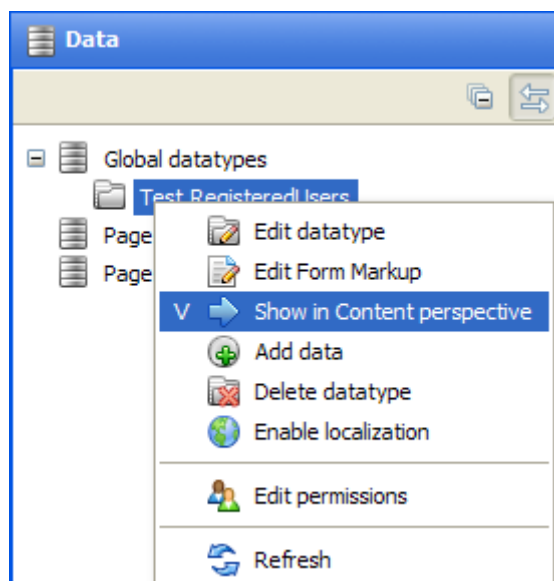


Figure 8: Indication of the datatype available in Content perspective

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

## 2.5 Adding Data Items

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

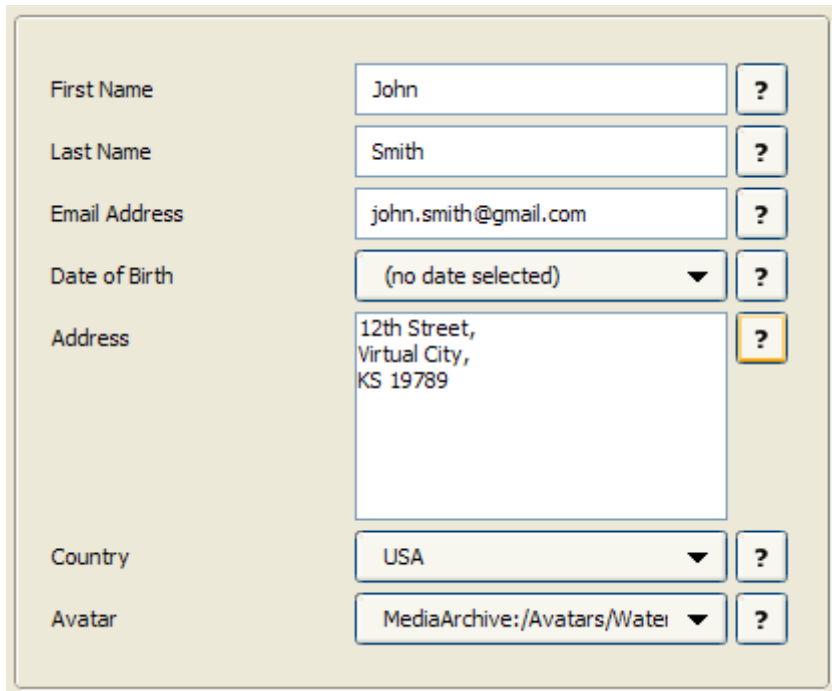
You can fill the global datatype by adding data items to it. When adding a data item, you enter values in the fields added to this datatype. Each datatype has its own set of fields.

You can add data items not only in the Data perspective but also in the Content perspective if you make the datatype 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 datatypes with data from their web browsers.

**To add a data item:**

1. In the Data Navigator, locate and select the datatype you want to add a data item to.
2. Click **Add Data Item** on the toolbar. The datatype item editor opens in the working area.
3. Fill out the datatype fields.



First Name	John	?
Last Name	Smith	?
Email Address	john.smith@gmail.com	?
Date of Birth	(no date selected)	?
Address	12th Street, Virtual City, KS 19789	?
Country	USA	?
Avatar	MediaArchive:/Avatars/Water	?

Figure 9: Adding a data item

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

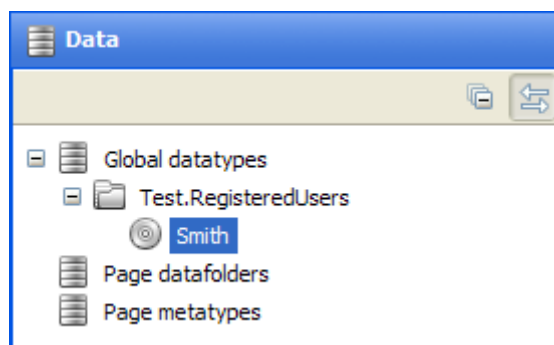


Figure 10: A data item added

### 3 Creating and Managing Fields

Each datatype might have one or more fields. You should add at least one field when creating a datatype. 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 datatype, 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.

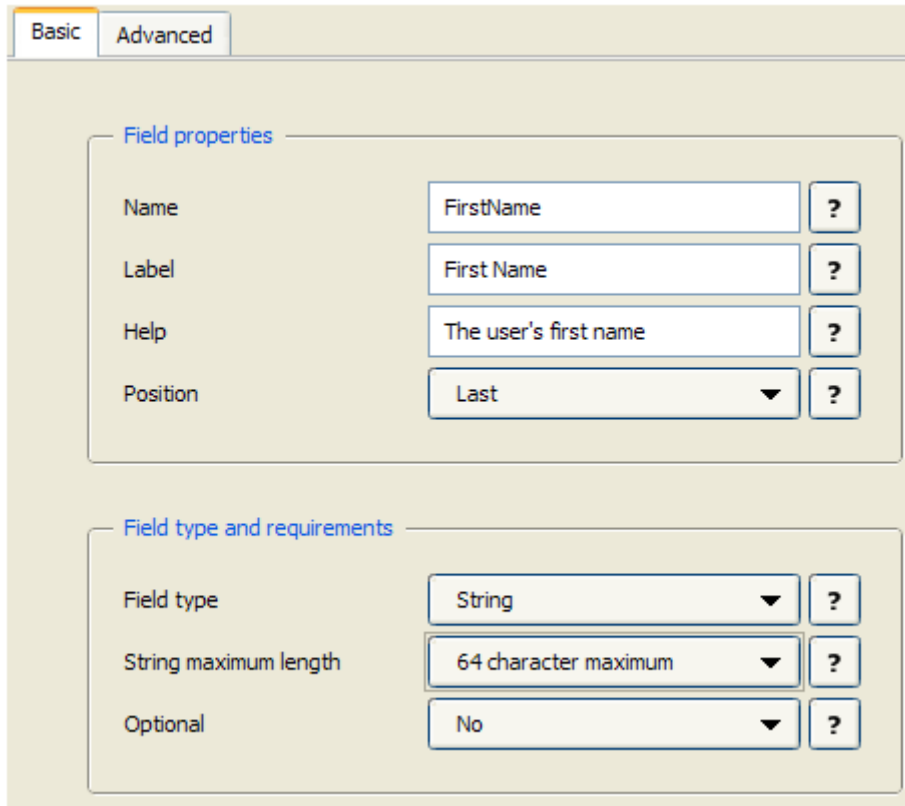
Each field you add should be of a certain type. The type determines what kind of data the field can hold. Based on the field type, some of the field's properties may differ. For example, for a string type you should specify the string's maximum length, for a decimal type – the number of decimal places, 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 datatype with a closer look at field types.

### 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.



The screenshot shows a configuration window with two tabs: 'Basic' (selected) and 'Advanced'. Under the 'Basic' tab, there are two main sections:

- Field properties:** Contains four rows, each with a label, a text input field, and a help button (question mark icon).
  - Name: Input field contains 'FirstName'.
  - Label: Input field contains 'First Name'.
  - Help: Input field contains 'The user's first name'.
  - Position: A dropdown menu showing 'Last'.
- Field type and requirements:** Contains three rows, each with a label, a dropdown menu, and a help button (question mark icon).
  - Field type: Dropdown menu showing 'String'.
  - String maximum length: Dropdown menu showing '64 character maximum'.
  - Optional: Dropdown menu showing 'No'.

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 datatype or on the forms based on these datatypes. 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.

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

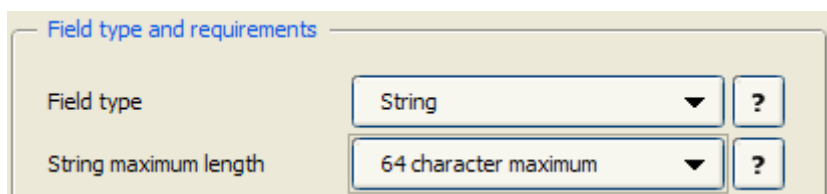
For each global datatype, C1 allows you to create fields of the following types:

- String
- 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*.



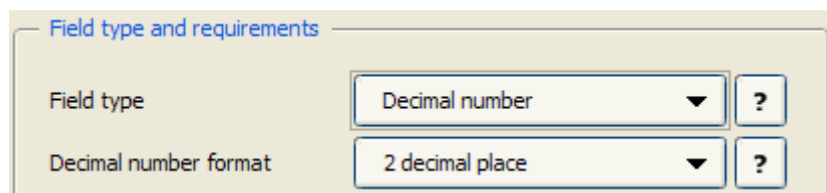
Field type and requirements

Field type: String

String maximum length: 64 character maximum

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*.



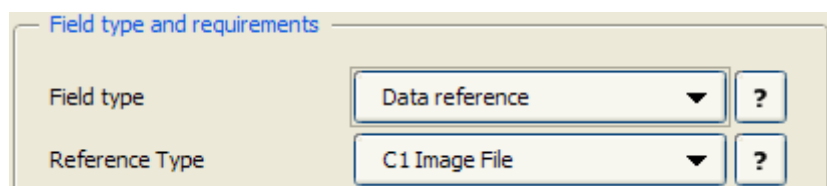
Field type and requirements

Field type: Decimal number

Decimal number format: 2 decimal place

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.



Field type and requirements

Field type: Data reference

Reference Type: C1 Image File

Figure 14: Additional property for the Data reference field

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 datatypes, you can also select them for the reference type. In this case, the data items of this datatype 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 datatype.

The screenshot shows the 'Advanced' tab of a configuration dialog. It contains four sections, each with a title and a control:

- Form field presentation:** 'Widget type' is set to 'TextBox'.
- Field validation:** 'Validation rules' is set to 'Add validation rules...'.
- Structural presentation:** 'Is title field' is checked with the label 'Use this as title field in lists'. 'Field grouping' is set to 'No grouping'.
- Field default value:** 'Default value' is set to '(no default value)'.

Figure 15: Advanced field properties

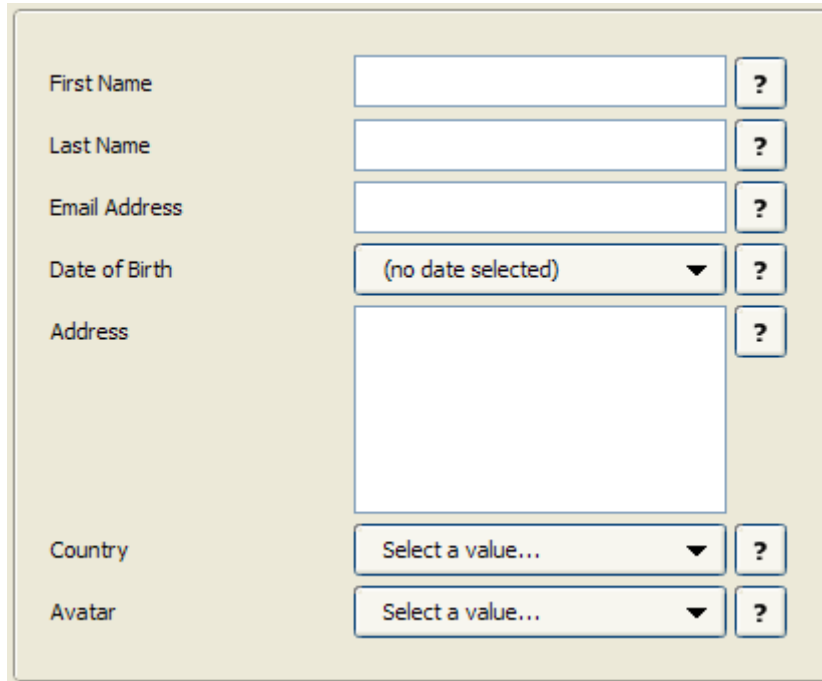
Every time you add another field, its advanced properties are set to defaults automatically. You can choose to override these default values.

Field Types	
Property	Description
Widget type	A widget to serve as an input control on a form representing this datatype.
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.
Field grouping	When the <i>Group by this field</i> 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 datatype, you can also select this field to <i>Group as 2<sup>nd</sup> priority</i> and so on.
Default value	The value, to which the field will be set by default when a new data item is added.



## 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 datatype 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.



First Name	<input type="text"/>	?
Last Name	<input type="text"/>	?
Email Address	<input type="text"/>	?
Date of Birth	<input type="text" value="(no date selected)"/>	?
Address	<input type="text"/>	?
Country	<input type="text" value="Select a value..."/>	?
Avatar	<input type="text" value="Select a value..."/>	?

Figure 16: Widgets

When you add fields to a datatype, 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 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.

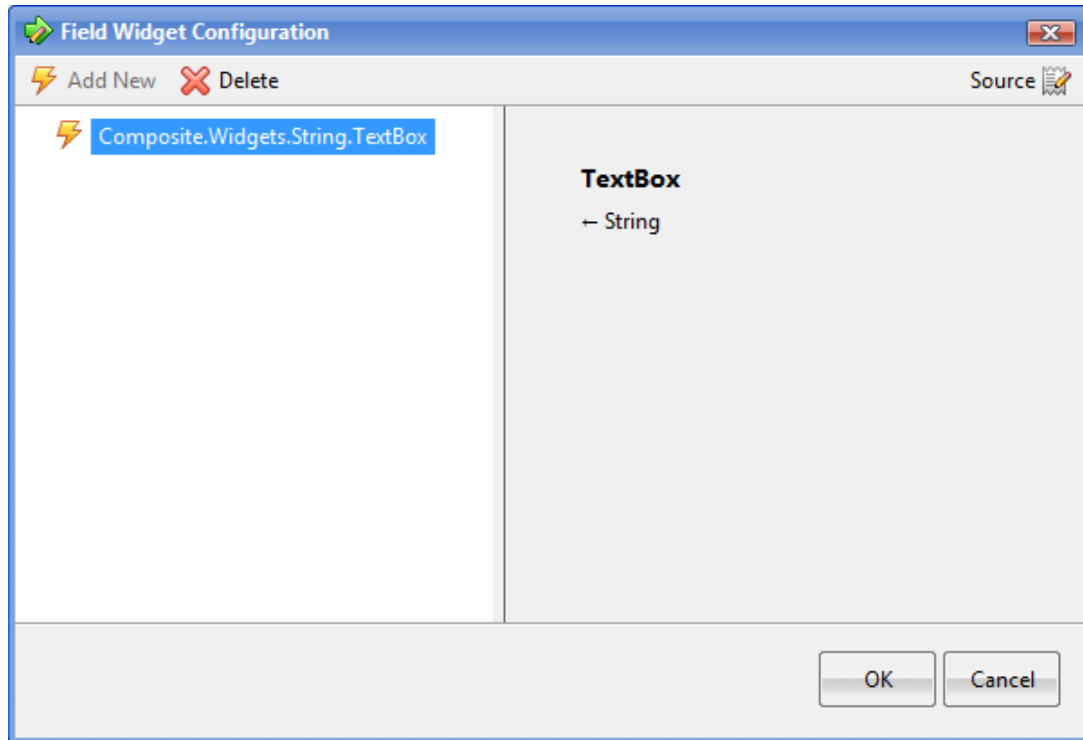


Figure 17: Currently selected widget

3. Select the current widget and click **Delete**. This will enable the **Add New** button.
4. Click **Add New** to open the **Select Widget** window. Here, you can see all the widgets available for this specific field.

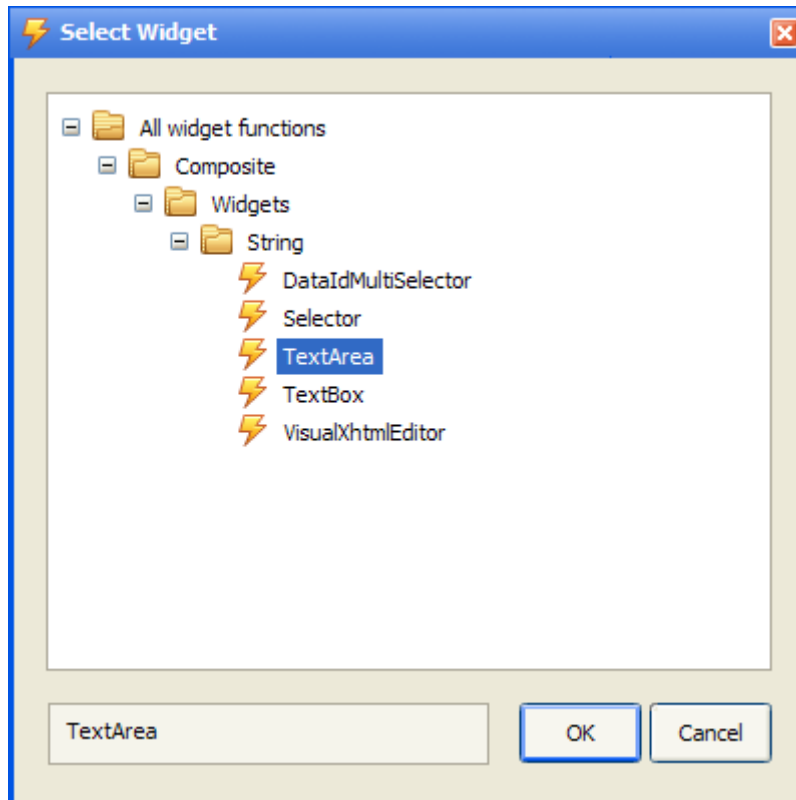


Figure 18: Selecting a different widget

5. Locate and select the widget you want to set for the field and click **OK**.
6. Click **OK** in the **Field Widget Configuration** window.

The **Widget type** field is now set to the widget you have selected.

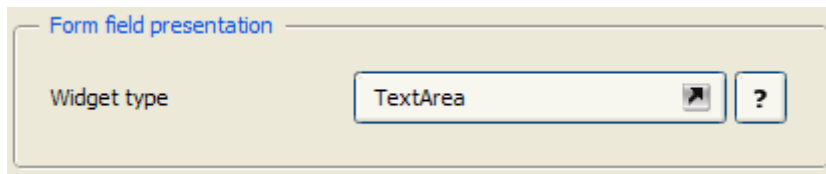


Figure 19: Newly selected widget

## 4.2 Overview of C1 Widgets

C1 features a number of widgets that can be used with one or more datatype 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.

#### 4.2.1 `TextBox`

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



Figure 20: `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.

#### 4.2.2 TextArea

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



Figure 21: TextArea

This widget is only available for the String field:

- `Composite.Widgets.String.TextArea`

#### 4.2.3 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.

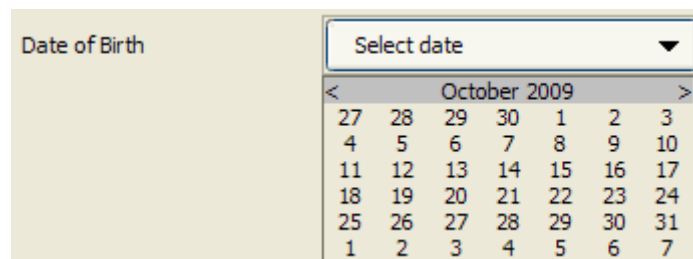


Figure 22: DateSelector

This widget is only available for the Date field:

- `Composite.Widgets.Date.DateSelector`

The DateSelector widget is default for the Date field.

#### 4.2.4 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 click the widget again and get a list of predefined times to choose from

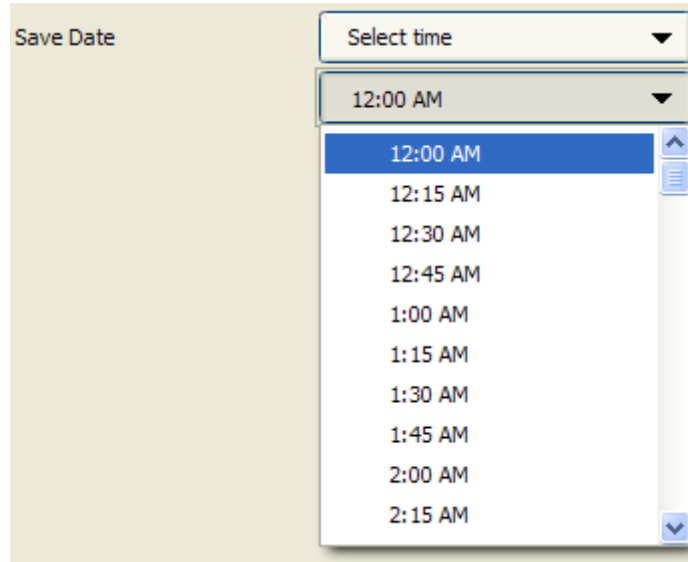


Figure 23: DateTimeSelector

This widget is only available for the Date field:

- `Composite.Widgets.Date.DateTimeSelector`

#### 4.2.5 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.

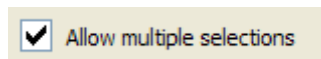


Figure 24: CheckBox

This widget is only available for the Boolean field:

- `Composite.Widgets.Bool.CheckBox`

The CheckBox widget is default for the Boolean field.

#### 4.2.6 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 25: BoolSelector

This widget is only available for the Boolean field:

- Composite.Widgets.Bool.BoolSelector

#### 4.2.7 VisualXhtmlEditor

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

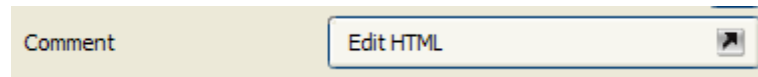


Figure 26: VisualXhtmlEditor

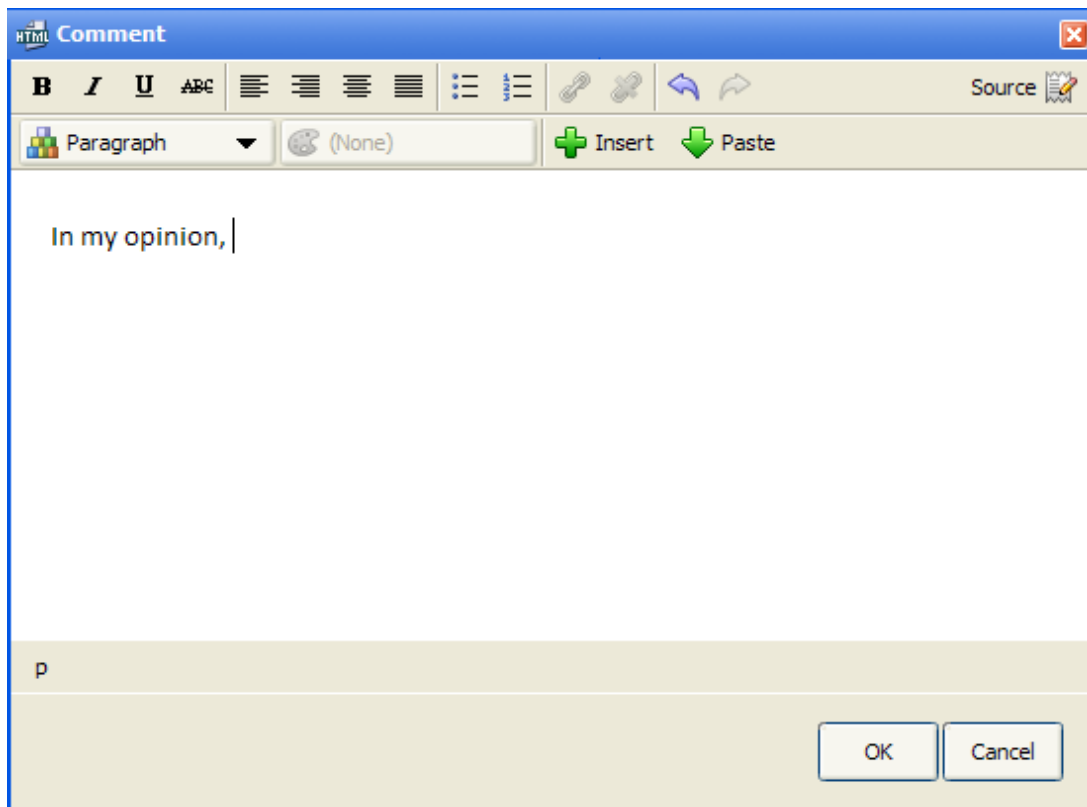


Figure 27: Launched Visual XHTML Editor

This widget is only available for the string field:

- `Composite.Widgets.String.VisualXhtmlEditor`

#### 4.2.8 `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 datatypes, 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.

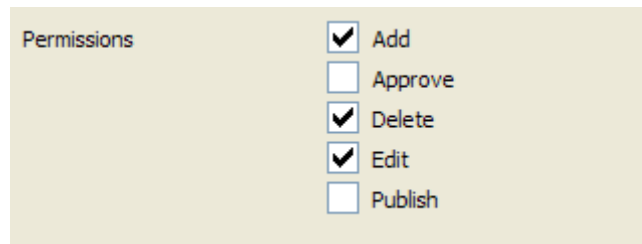


Figure 28: `DataIdMultiSelector`: Verbose UI View

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.

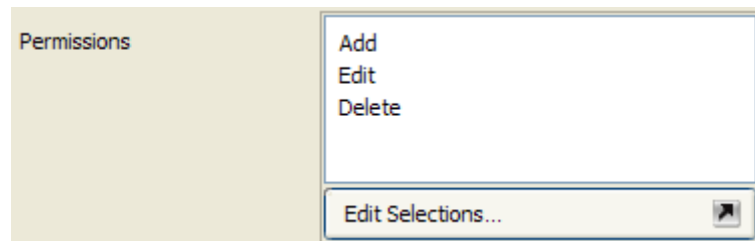


Figure 29: `DataIdMultiSelector`: Compact UI View



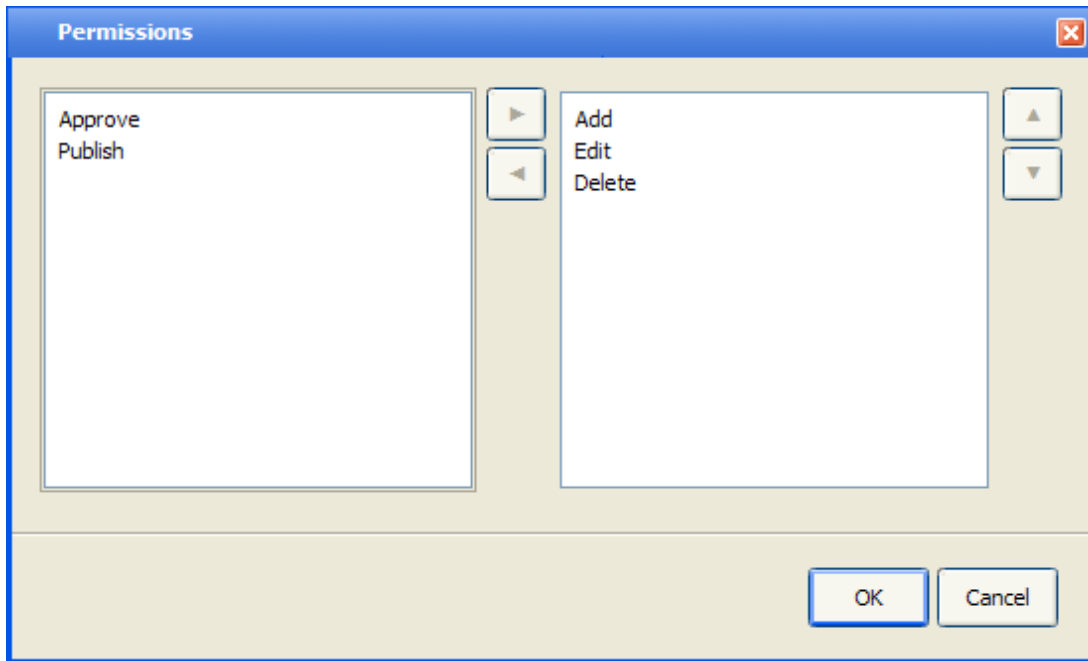


Figure 30: DataIdMultiSelector: Selecting values

This widget is only available for the string field:

- `Composite.Widgets.String.DataIdMultiSelector`

#### 4.2.9 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.

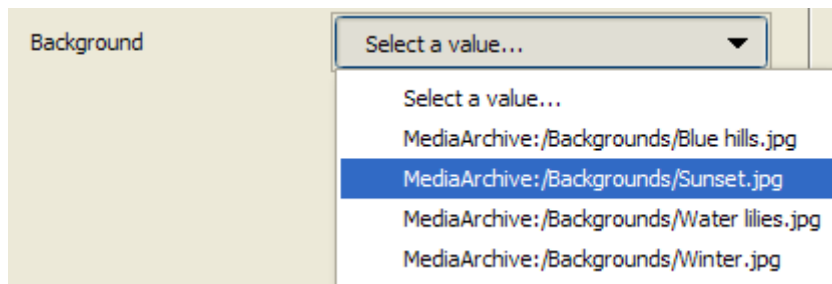


Figure 31: Selector

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

- `Composite.Widgets.String.Selector`
- `Composite.Widgets.CompositeDataTypesImageFile.Selector`
- `Composite.Widgets.CompositeDataTypesImageFile.Selector`
- `Composite.Widgets.CompositeDataTypesImageFileFolder.Selector`
- `Composite.Widgets.CompositeDataTypesIPage.Selector`

The Selector widget is default for the Data Reference fields of C1 Page and datatype reference types.

#### 4.2.10 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.

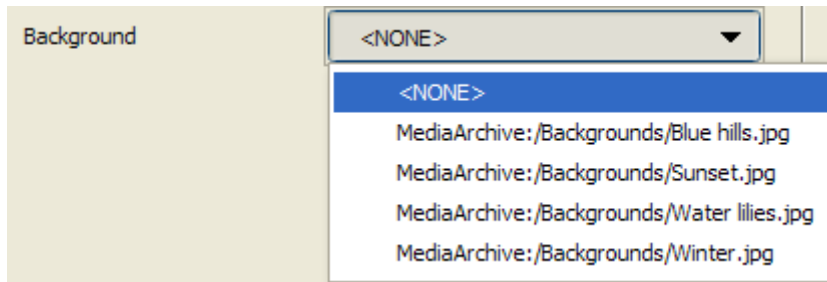


Figure 32: OptionalSelector

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

- `Composite.Widgets.String.OptionalSelector`
- `Composite.Widgets.CompositeDataTypesImageFile.OptionalSelector`
- `Composite.Widgets.CompositeDataTypesImageFileFolder.OptionalSelector`
- `Composite.Widgets.CompositeDataTypesImageFileFolder.OptionalSelector`
- `Composite.Widgets.CompositeDataTypesPage.OptionalSelector`

#### 4.2.11 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.

Please note that you cannot use these widgets with `Composite.Forms.Renderer`, in which case you should replace them with the corresponding Selectors or Optional Selectors. (Please see [Using Input Forms with Datatypes](#) for more information.)

## 5 Creating Specific Fields

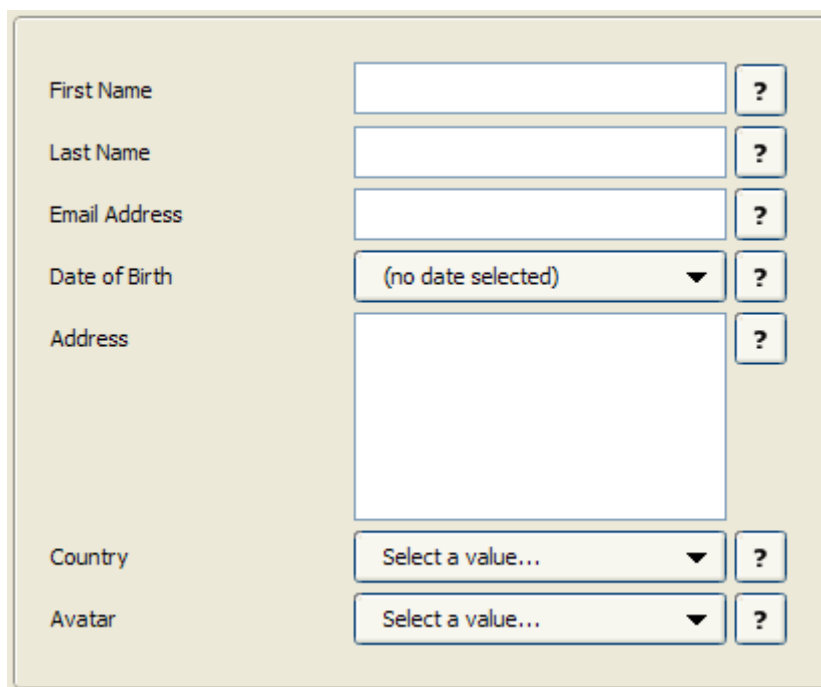
Now that you have learned how to create datatypes and add fields to these datatypes, set properties and select widgets for the fields, let's see how to create fields that will serve specific purposes.

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 datatype 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.



The image shows a form with the following fields and widgets:

- First Name:** A standard text input field.
- Last Name:** A standard text input field.
- Email Address:** A standard text input field.
- Date of Birth:** A date selection dropdown menu showing "(no date selected)".
- Address:** A large, multi-line text area.
- Country:** A dropdown menu with the text "Select a value...".
- Avatar:** A dropdown menu with the text "Select a value...".

Each field is accompanied by a small square button containing a question mark, likely representing a help or tooltip function.

Figure 33: 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 datatype.
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.

The image shows a configuration window for a 'Simple Text Field' with two tabs: 'Basic' and 'Advanced'. The 'Basic' tab is active and contains two sections: 'Field properties' and 'Field type and requirements'. Each property has a text input field, a dropdown menu, or a radio button, and a help icon (question mark).

Property	Value	Help
Name	FirstName	?
Label	First Name	?
Help	The user's first name	?
Position	1.	?
Field type	String	?
String maximum length	32 character maximum	?
Optional	No	?

Figure 34: Simple Text Field: basic properties

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

6. If necessary, specify other advanced properties.

The screenshot shows the 'Advanced' tab of a configuration window. It contains four sections:

- Form field presentation:** 'Widget type' is set to 'TextBox'.
- Field validation:** 'Validation rules' is set to 'Add validation rules...'.
- Structural presentation:** 'Is title field' is an unchecked checkbox labeled 'Use this as title field in lists'. 'Field grouping' is set to 'No grouping'.
- Field default value:** 'Default value' is set to '(no default value)'.

Figure 35: 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 datatype.
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*.

- Specify whether this field is optional or required.

Field properties	
Name	Address ?
Label	Address ?
Help	The user's address ?
Position	5. ?

Field type and requirements	
Field type	String ?
String maximum length	Unlimited length ?
Optional	No ?

Figure 36: Large Text Field: basic properties

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

Field Widget Configuration

Add New Delete Source

Composite.Widgets.String.TextBox

**TextBox**  
← String

OK Cancel

Figure 37: Field Widget Configuration window

8. Select **Composite.Widgets.String.TextBox** and click **Delete**.
9. Click **Add New** to open the **Select Widget** window and locate and select the **TextArea** widget (**Composite.Widgets.String.TextArea**).

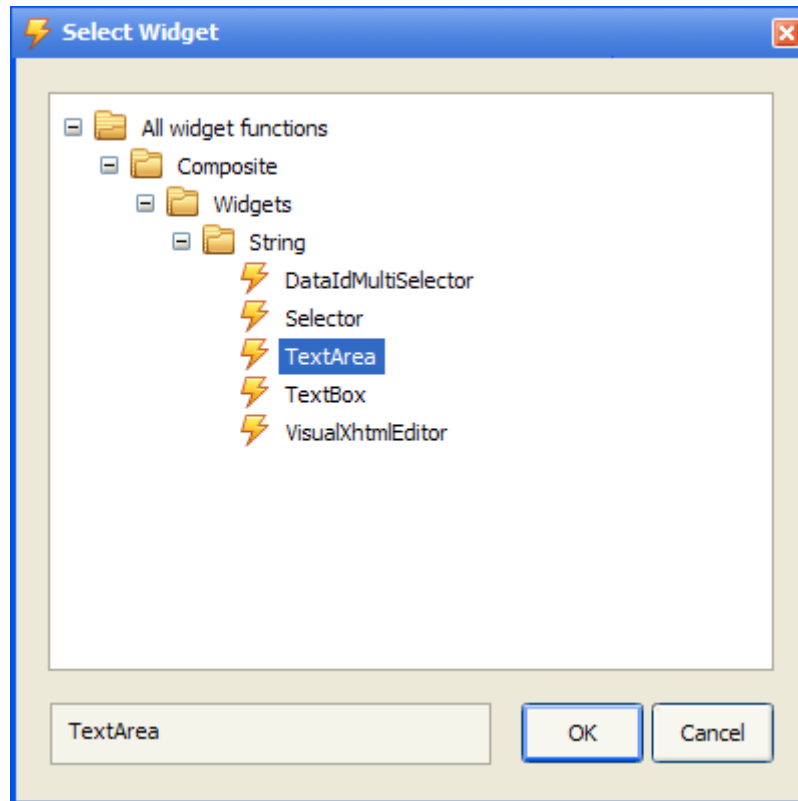


Figure 38: Selecting TextArea widget

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

11. If necessary, specify other advanced properties.

The image shows a configuration window for a 'Large Text Field' with the 'Advanced' tab selected. The window is divided into four sections, each with a title and a set of controls:

- Form field presentation:** 'Widget type' is set to 'TextArea'.
- Field validation:** 'Validation rules' is set to 'Add validation rules...'.
- Structural presentation:** 'Is title field' is an unchecked checkbox with the text 'Use this as title field in lists'. 'Field grouping' is set to 'No grouping'.
- Field default value:** 'Default value' is set to '(no default value)'.

Figure 39: Large Text Field: advanced properties

12. 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 datatype.
2. Specify its basic properties such as its *name*, *label*, *help text* and *position*.
3. For the **Field type**, select *Date*.



- Specify whether this field is optional or required.

The image shows a configuration window with two tabs: 'Basic' and 'Advanced'. The 'Basic' tab is active. It contains two sections: 'Field properties' and 'Field type and requirements'. Each section has four rows of configuration options, each with a help icon (a question mark in a square).

Section	Property	Value	Help Icon
Field properties	Name	DOB	?
	Label	Date of Birth	?
	Help	The user's date of birth	?
	Position	4.	?
Field type and requirements	Field type	Date	?
	Optional	No	?

Figure 40: Date Field: basic properties

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

6. If necessary, specify other advanced properties.

The screenshot shows the 'Advanced' tab of a configuration interface. It contains four sections:

- Form field presentation:** 'Widget type' is set to 'DateSelector'.
- Field validation:** 'Validation rules' is set to 'Add validation rules...'.
- Structural presentation:** 'Is title field' is an unchecked checkbox labeled 'Use this as title field in lists'. 'Field grouping' is set to 'No grouping'.
- Field default value:** 'Default value' is set to '(no default value)'.

Figure 41: 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 datatype.
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*.

- Specify whether this field is optional or required.

Field properties	
Name	Avatar ?
Label	Avatar ?
Help	The user's avatar ?
Position	7. ▼ ?

Field type and requirements	
Field type	Data reference ▼ ?
Reference Type	C1 Image File ▼ ?
Optional	No ▼ ?

Figure 42: Data Reference Field of C1 Image type: basic properties

- 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 Datatypes](#) for more information.)

6. If necessary, specify other advanced properties.

The screenshot shows the 'Advanced' tab of a configuration interface. It contains four sections, each with a title and a control:

- Form field presentation:** 'Widget type' is set to 'ImageSelector'.
- Field validation:** 'Validation rules' is set to 'Add validation rules...'.
- Structural presentation:** 'Is title field' has an unchecked checkbox and the text 'Use this as title field in lists'. 'Field grouping' is set to 'No grouping'.
- Field default value:** 'Default value' is set to '(no default value)'.

Figure 43: 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 datatypes.

## 5.5 How to Create String Field with Selector

In some cases, you do not need to create a separate datatype, 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 datatype.
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*.

- Specify whether this field is optional or required.

Field properties	
Name	Country ?
Label	Country ?
Help	The user's country ?
Position	6. ▼ ?

Field type and requirements	
Field type	String ▼ ?
String maximum length	Unlimited length ▼ ?
Optional	No ▼ ?

Figure 44: The String field: basic properties

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

- Click **Add New** to open the **Select Widget** window and locate and select the *Selector* widget. Click **OK**.

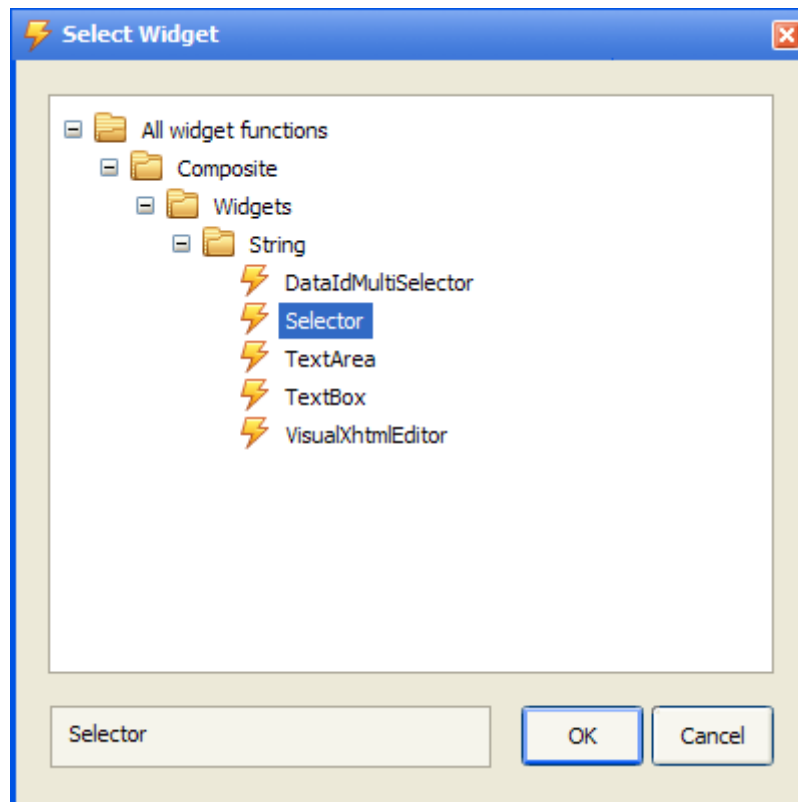


Figure 45: Selecting the Selector widget

- Select the **Options** parameter of the **Selector** widget and select *Execute a function call* for its **Value type**.

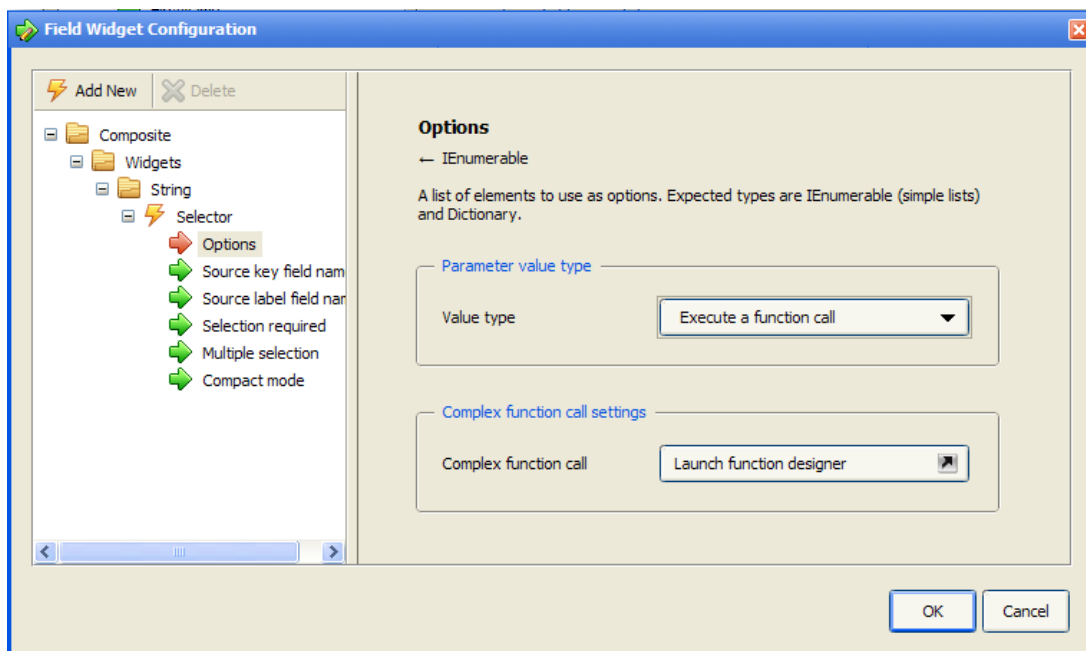


Figure 46: Setting the Options parameter

- Click *Launch function designer* and locate and select the **Composite.Utills.String.Split** function in the **Select Function** window. Click **OK**.

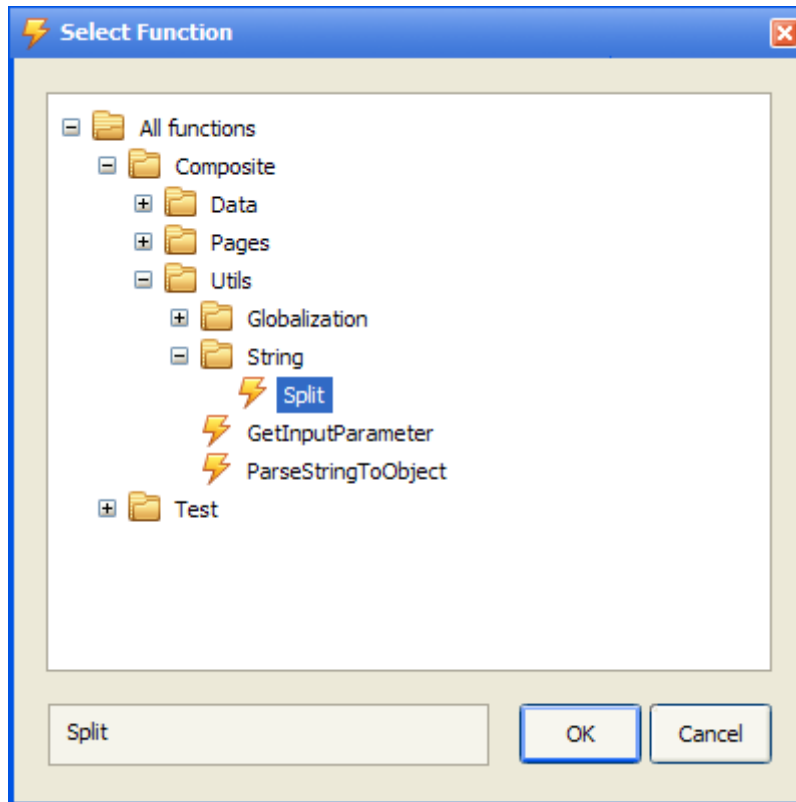


Figure 47: Selecting the **Composite.Utills.String.Split** function

- 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**.

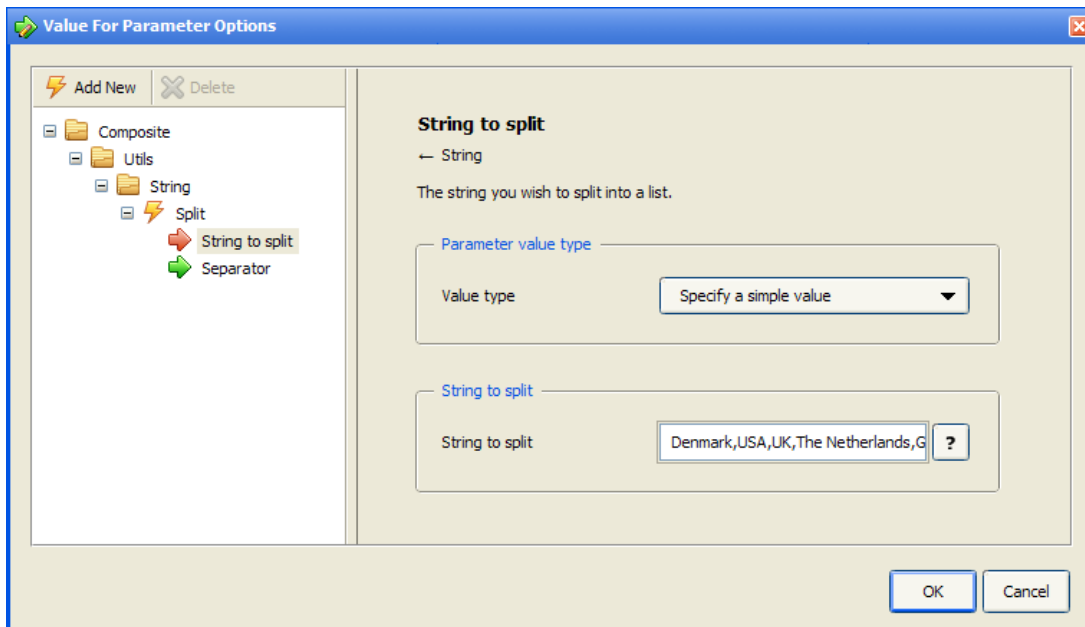


Figure 48: Specifying a string to split

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

11. If necessary, specify other advanced properties.

The screenshot shows the 'Advanced' tab of a configuration interface. It is organized into four distinct sections, each with a title and a set of controls:

- Form field presentation:** Contains a 'Widget type' dropdown menu set to 'Selector' and a help button (?).
- Field validation:** Contains a 'Validation rules' dropdown menu set to 'Add validation rules...' and a help button (?).
- Structural presentation:** Contains two controls: 'Is title field' with an unchecked checkbox and the text 'Use this as title field in lists' (with a help button ?), and 'Field grouping' with a dropdown menu set to 'No grouping' (with a help button ?).
- Field default value:** Contains a 'Default value' dropdown menu set to '(no default value)' and a help button (?).

Figure 49: 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 Datatypes

To allow the visitors of your website to add data to your datatypes 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 datatype that will hold the input information.

The data added via these forms will be added to the corresponding datatype 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 datatype* available to hold data entered by the user via the form with all necessary fields added and set up.

Please note that if your datatype 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.Widget.DataReference.DataTypesImageFile.Selector`
- C1 Page: `Composite.Widget.DataReference.DataTypesIPage.Selector`
- C1 Media File: `Composite.Widget.DataReference.DataTypesIMediaFile.Selector`
- C1 Media Folder:  
`Composite.Widget.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 datatype that will hold the input data for this form. All the fields in this datatype will become available in the form via the widgets you selected for this datatype.

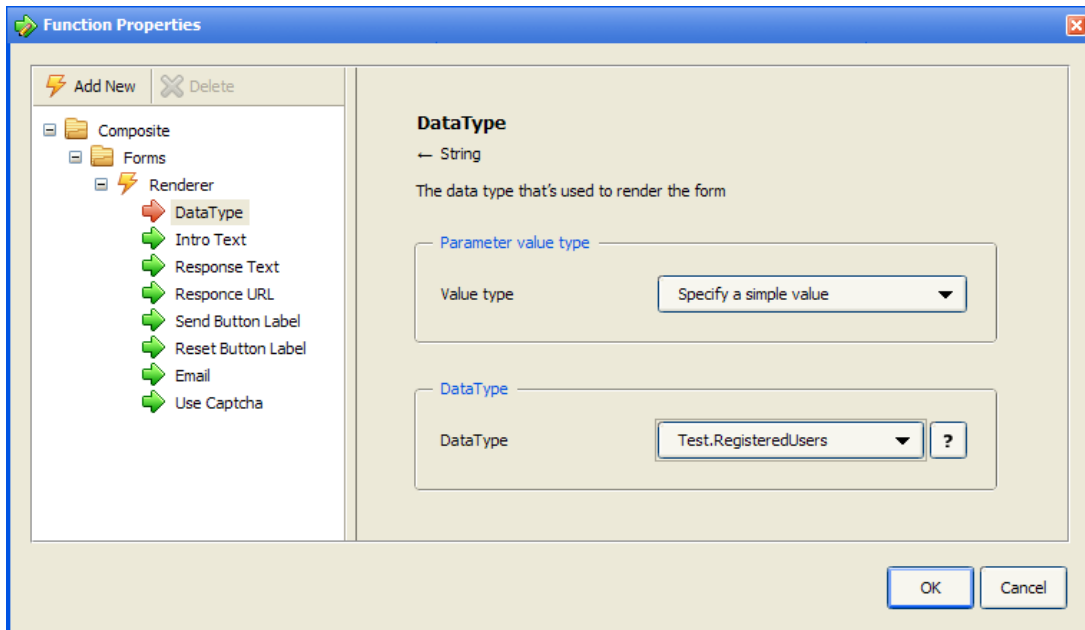


Figure 50: Setting up `Composite.Forms.Renderer`

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 datatype.

First Name	<input type="text" value="John"/>
Last Name	<input type="text" value="Smith"/>
Email Address	<input type="text" value="john.smith@gmail.com"/>
Date of Birth	<input type="text" value="1/1/1975"/>
Address	<input style="width: 100%; height: 100%;" type="text" value="12th St,&lt;br/&gt;Virtual City,&lt;br/&gt;KS 18789"/>
Country	<input type="text" value="USA"/>
Avatar	<input type="text" value="MediaArchive:/Avatars/W:"/>
<input type="button" value="Submit"/>	

Figure 51: 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 datatype 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 then 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.

**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 then 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.

## 7 Test Your Knowledge

### 7.1 TASK 1

1. Create a global datatype.
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* datatype 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 datatype 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.Widget.DataReference.DataTypesImageFile.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 datatype.
5. Make the datatype 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* datatype.
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 datatype from the Content perspective.
2. Open the *Registration* page in a web browser.
3. Fill out and submit the form.