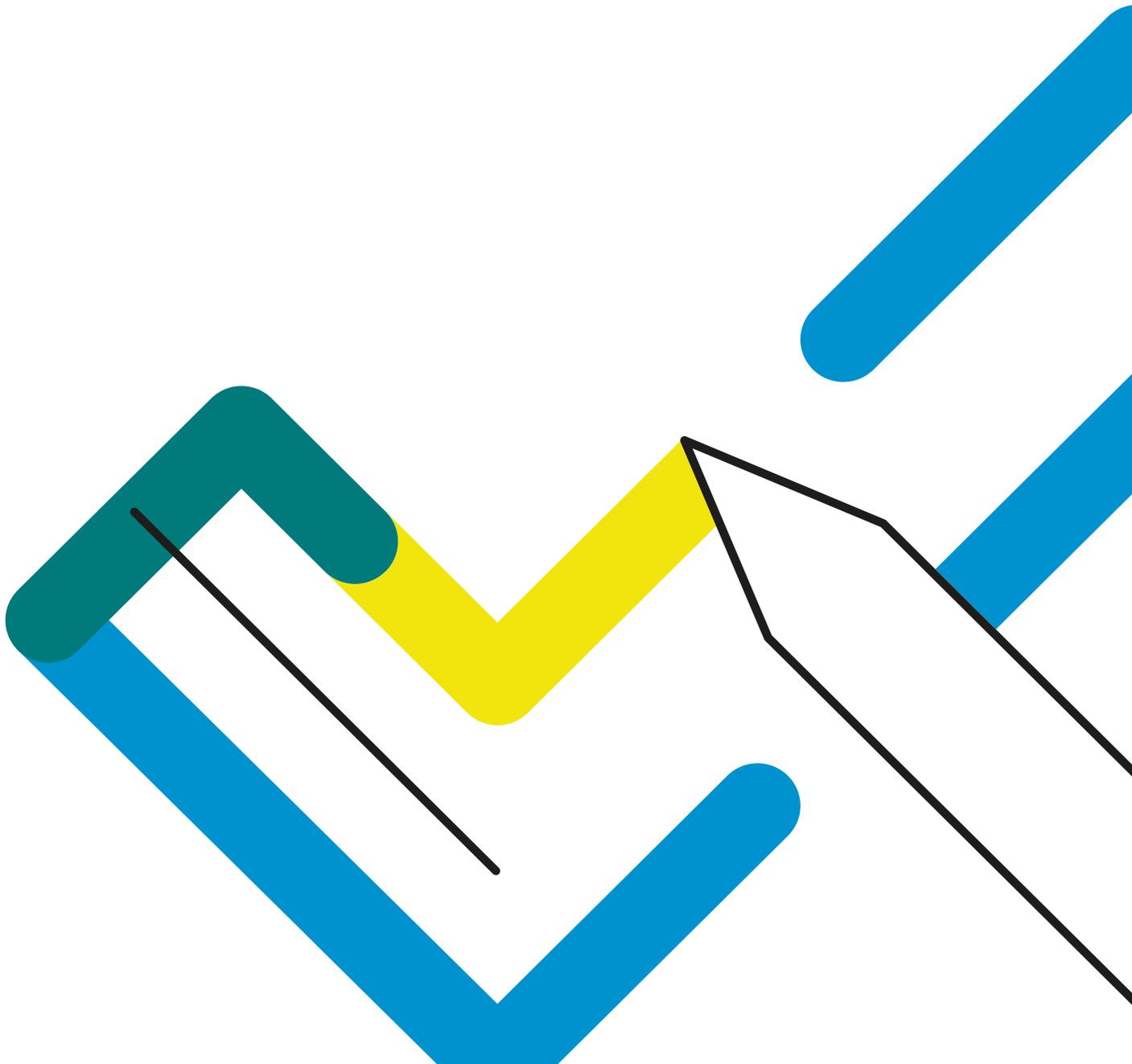


User Manual

Version 6.5.6



Monday Webforms for FirstSpirit: User Manual

Copyright © 2023 Formcentric GmbH
Breite Str. 61, 22767 Hamburg
Germany

The contents of this document – whether in whole or in part – may not be reproduced, conveyed, disseminated or stored in any form whatsoever without obtaining prior written permission from Formcentric GmbH.

Disclaimer

We reserve the right to alter the software and the contents of the manual without notice. We accept no liability for the accuracy of the contents of the manual, nor for any losses that may result from the use of this software.

Trademarks

In the course of this manual, references may be made to trademarks that are not explicitly marked as such. Even if such a mark is not given, the reader should not conclude that the name is free of third-party rights.

Please note: Printed copies are uncontrolled when printed.

1. Introduction	1
1.1. Terminology	1
2. Overview	2
2.1. Function overview	2
3. Form editing interface	4
3.1. On-the-fly validation	5
3.2. Form elements	6
3.2.1. Form	6
3.2.2. Input Field	13
3.2.3. Password field	21
3.2.4. Textarea	23
3.2.5. Single Choice	25
3.2.6. Multiple Choices	26
3.2.7. List of Choices	28
3.2.8. Option	29
3.2.9. Hidden Field	30
3.2.10. Upload field	32
3.2.11. Condition	33
3.2.12. Page Break	36
3.2.13. Page break condition	38
3.2.14. Calculated Value	39
3.2.15. Paragraph	40
3.2.16. Captcha	41
3.2.17. Button	41
3.2.18. Fieldset	42
3.2.19. Layout	43
3.2.20. Summary	44
3.3. Variables	46
3.4. Actions	47
3.4.1. Send as Email	48
3.4.2. Excel File	54
3.4.3. PDF	55
3.4.4. Data source	57
3.4.5. Media Management	58
3.4.6. Webforms Analytics	60
3.4.7. Redirect	60
3.4.8. Webhook	61
3.4.9. Sequence	63
3.5. Markdown	63
A. Reserved identifiers	65

1. Introduction

This manual describes how to use Monday Webforms. Monday Webforms is a form manager extension for the FirstSpirit CMS.

This manual will show you how to use Webforms to create and edit forms in the FirstSpirit SiteArchitect.

1.1. Terminology

This manual makes use of the following terms:

Term	Description
Form	An HTML web form that is displayed in a browser.
Form elements	All of the elements used when constructing a form (input fields, drop-down lists, check boxes, captchas, etc.).
Form data	The data items that are entered into a form by the user.
Form author	The person that creates and edit forms.
User	The person that fills out a form.
Editor	The FirstSpirit JavaClient
Frontend	The web page that is created with the FirstSpirit CMS.

2. Overview

You use the Form Editor to create and edit any web forms that you need within the FirstSpirit SiteArchitect. The Preview always shows you the current state of the form you are working on. This also gives you the chance to test the form as part of the creation process.

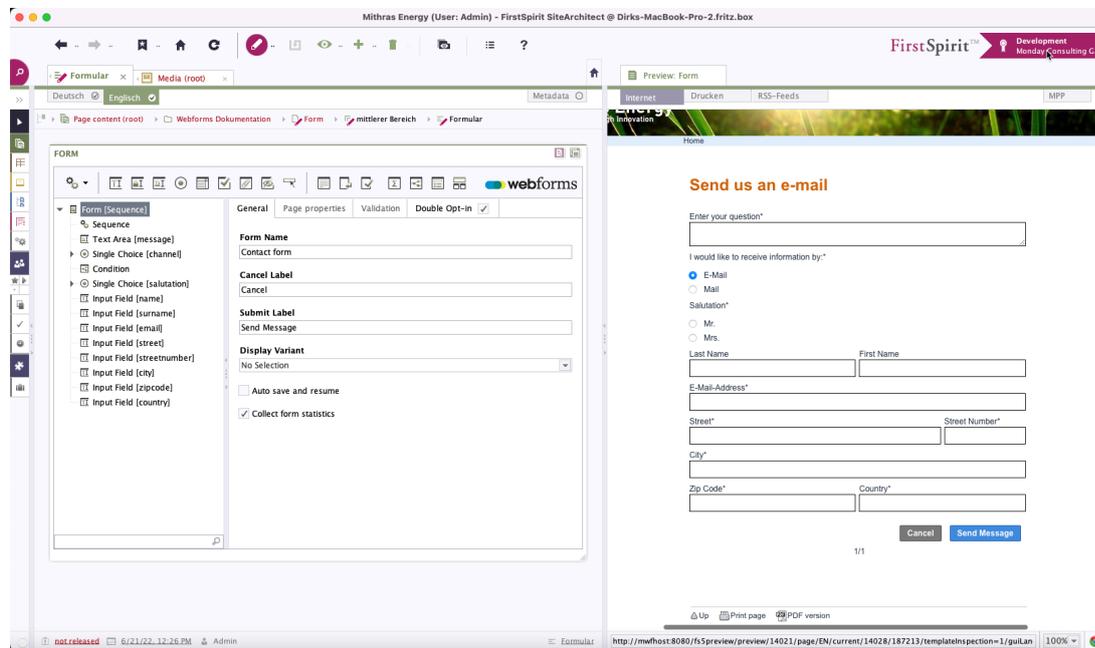


Figure 2.1. Monday Webforms in the JavaClient

2.1. Function overview

Form elements: Monday Webforms supports all of the form elements defined in the HTML standard, such as input fields, drop-down lists or radio buttons. It also provides you with other elements such as *Captcha*, *Calculated Value* or *Summary*.

Multi-page forms: Detailed or complex forms can be split up over multiple form pages. Form users can then page back and forth between the individual form pages in order to change or add the data they have entered.

Conditions: Based on the input entered by the user, individual form elements – or even whole form pages – can be shown or hidden dynamically. To do so, you define the business rules in the Condition Editor beforehand.

Field validators: A range of field validators are available to you for validating user input. You can select these validators when creating a form field. You can modify most validators to suit specific requirements by using specialised attributes. As a rule, all validation of input takes place on the server, to avoid form data manipulation.

Default field values: You can give input fields default values, which can be fixed, variable or user-specific.

Freely-definable actions: By selecting a processing mode (action) in the Editor, you can decide how the form data will be processed after submission. Out of the box, Webforms

comes complete with actions for sending data by email and for outputting data to an Excel file. Other actions, customised to suit special requirements, can be developed with the help of an easy-to-use programming interface (API).

Security: Monday Webforms contains a security servlet filter as a safeguard against cross-site scripting (XSS) attacks and cross-site request forgery (XSRF) attacks. This filter removes illegal HTML tags, CSS and scripts from the form data submitted. The filter also checks to confirm that the form data contains a valid XSRF token.

3. Form editing interface

The interface that is used for working with the Form Editor is split into three areas:

- Toolbar [1]
- Tree view [2]
- Detail view [3]

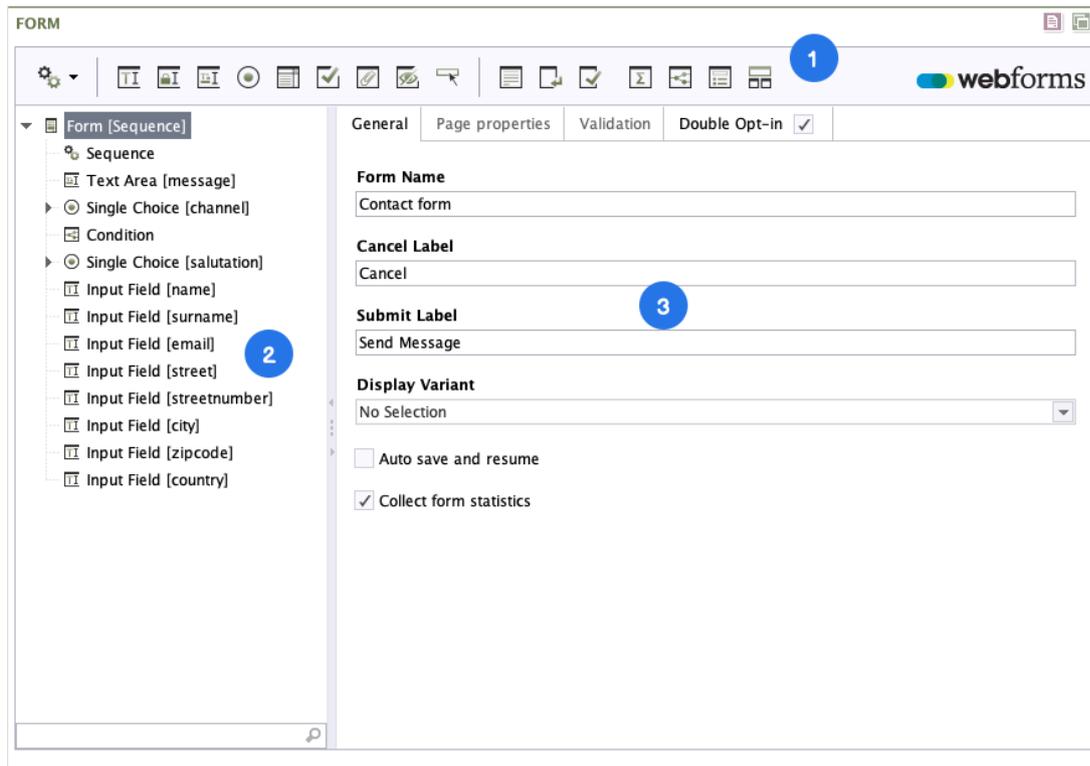


Figure 3.1. Form editing interface

On the toolbar [1], you will find all of the form elements that can be inserted below the element that is currently selected in the tree view [2]. By clicking a tool icon in the toolbar, you can add the corresponding element to the form. New elements are always inserted automatically below the currently selected element.

The tree view [2] shows you all of the elements of the form. By right-clicking on the element when selected, you can open its associated context menu. You can access the following functions via the context menu or the toolbar:

Insert new element ▶	
Move to top	Alt+Pos 1
Move up	Alt+Oben
Move down	Alt+Unten
Move to bottom	Alt+Ende
<hr/>	
 Copy	
 Paste	
 Duplicate	Alt+D
 Remove	Entf
<hr/>	
 Expand All	Alt+Plus
 Collapse All	Alt+Minus
 Show additional information	

Figure 3.2. Form Editor context menu

In the detail view [3], you can edit the properties of the form element currently selected in the tree.

3.1. On-the-fly validation

The Form Editor validates the form while it is being entered (on-the-fly) and marks incomplete elements or elements with errors appropriately both in the tree view and in the detail view.

Please note: You cannot save a form if it contains errors.

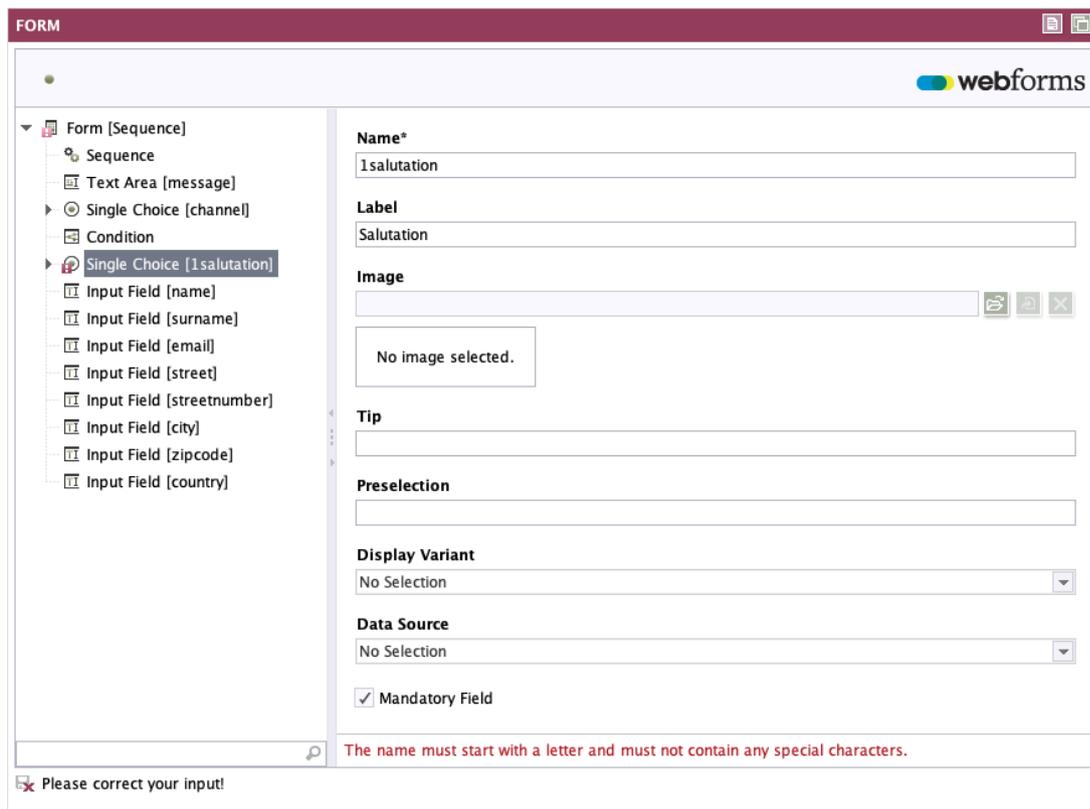


Figure 3.3. Marking of forms with errors

3.2. Form elements

When you use Monday Webforms to create your form, all of the form elements as described below are available as standard. These include both input elements such as the one-line text input field as well as read-only display elements such as the Summary. You can specify the labelling, presentation format and behaviour of a form element individually for each and every element.

You will need to enter a technical name for most of these form elements. This name is then used as a unique reference for the subsequent processing of the data that is input. For this reason, you cannot use the same name more than once within a single form. For technical reasons, the name must begin with a letter, must not contain any special characters and must not match a word in the list of reserved identifiers (Appendix A, *Reserved identifiers*).

3.2.1. Form

The *Form* element represents the form itself. You will always find this element at the top level of the tree view. The system arranges all of the other elements underneath this element.

For multi-page forms, the Form element corresponds to the first form page. All other pages are represented by an appropriate *Page Break* element (see Section 3.2.12, “Page Break”).

The form elements are output in the order in which they are displayed in the tree view.

The detail view of the form element is split over the tabs *General*, *Page properties*, *Validation* and *Double opt-in*. The various configuration options are explained in the following sections.

Properties tab

The screenshot shows the 'FORM' editor interface. On the left, a tree view shows a 'Form [Sequence]' element expanded to show its components: Sequence, Text Area [message], Single Choice [channel], Condition, Single Choice [salutation], Input Field [name], Input Field [surname], Input Field [email], Input Field [street], Input Field [streetnumber], Input Field [city], Input Field [zipcode], and Input Field [country]. The right pane is titled 'FORM' and has tabs for 'General', 'Page properties', 'Validation', and 'Double Opt-in' (checked). The 'General' tab is active and contains the following fields and options:

- Form Name:** Contact form
- Cancel Label:** Cancel
- Submit Label:** Send Message
- Display Variant:** No Selection
- Auto save and resume
- Collect form statistics

Figure 3.4. Form element – General tab

Form Name: Enter a descriptive name for the form here. This is used when displaying the form within Webforms Analytics.

Cancel Label: This text is displayed on the form’s “Cancel” button.

Submit Label: This text is displayed on the form’s “Submit” button.

Display Variant: Use this to pick an alternative presentation style for the form.

Auto save and resume: Activate this check box if you want to give the user the option of interrupting the form entry process and resuming it at a later point in time, without losing any of the data he or she has already entered. The input data is stored until the user has completed the form entry process and then submitted the form. This option is especially useful for multi-page forms.

Collect form statistics: If you activate this check box and have a Webforms licence that includes Webforms Analytics, then statistical metrics for this form will be collected and processed for reports. You can view these reports in the Webforms Analytics Reporting application and drill down further into reports on form user behaviour.

Page properties tab

You use the *Page Properties* tab to specify the properties for the first page of your form. The properties for other pages in your form can then be specified on the corresponding *Page Break* element (see Section 3.2.12, “Page Break”).

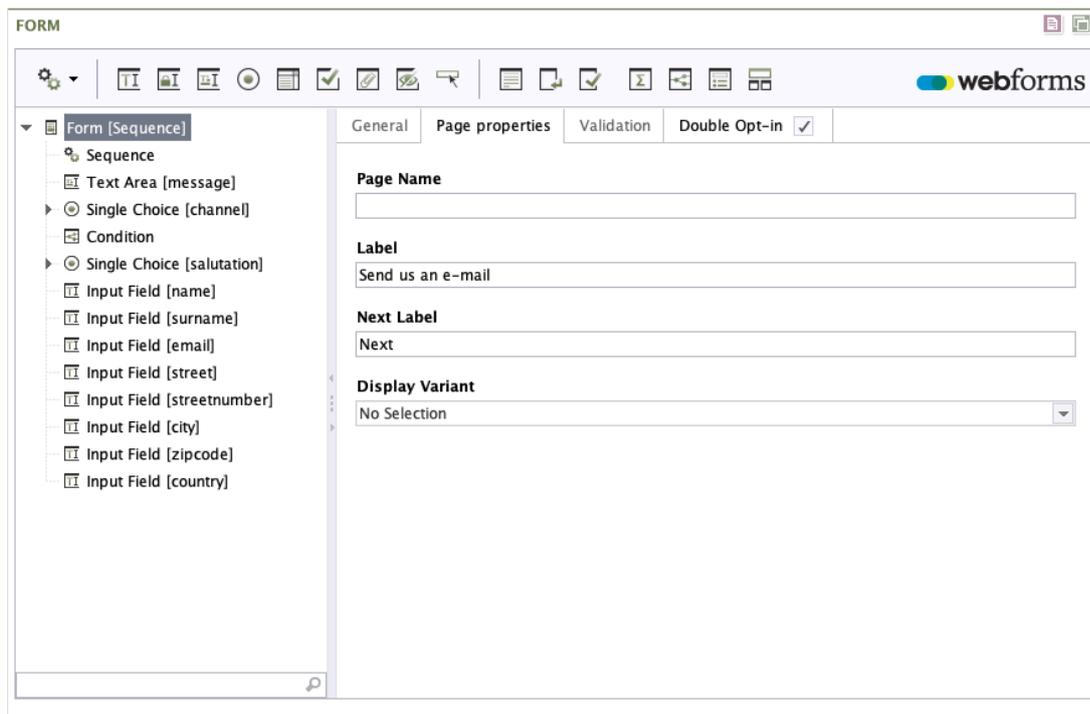


Figure 3.5. Form element – Page properties tab

Page Name: Enter the technical name of the first form page here.

Label: This text is output as a heading on the first page of the form.

Next Label: This text is displayed on the “Next” button that is shown on the first page of a multi-page form. This text item is not used for forms that have only one page.

Display Variant: Select an alternative display variant for the first page of the form.

Validation tab

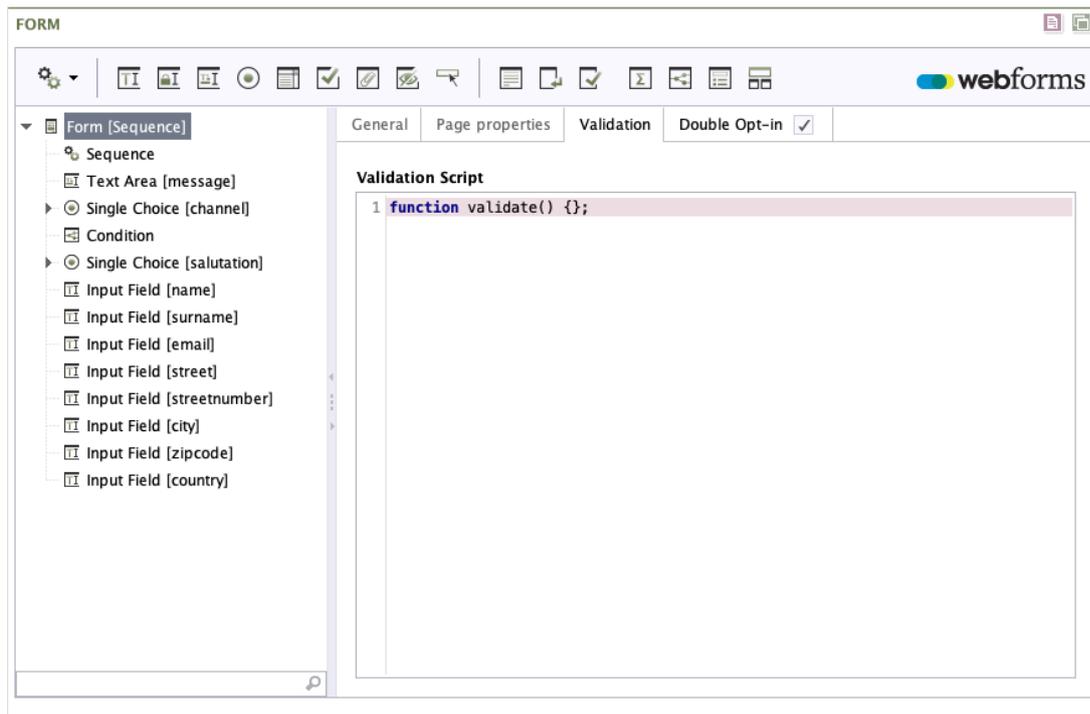


Figure 3.6. Form element – Validation tab

Validation Script: You can use the validation script to validate the data entered into the form. Unlike field validators, which verify only the value of a single field, this function gives you the opportunity to simultaneously check the values of multiple fields and to create dependencies between them.

When a form is first created, the field *Validation script* contains the empty method `validate()`. This is a JavaScript function that is executed every time the form is submitted. You can input your own validation logic into this function.

Please note: if the form data contains errors, the function must return a statement that clearly describes the error. If no error was determined during validation, then the script must return an empty string (`""`). In this case, the form data is considered to be correct and is sent for processing.

When creating the validation script, you can use all of the operations and functions available in JavaScript. For handling date values and drop-down lists, the functions `parseDate`, `parseAge`, `isEmptyList` and `isSelected` are available.

The function `parseDate(date format, date)` converts a character string into a JavaScript object of the *Datatype*. For the first parameter, specify the underlying date format; for the second parameter, give the date value.

```
parseDate( "dd/MM/yyyy" , "18/12/1969" );
```

The function `parseAge(dateFormat, date)` calculates a person's age based on their birth date. For the first parameter, specify the underlying date format; for the second parameter, give the date value.

```
parseAge("dd/MM/yyyy", "18/12/1969");
```

You can use the function *isEmptyList(selection)* to verify whether or not the user has selected anything for the specified selection.

```
isEmptyList(newsletter);
```

You can use the function *isSelected(selection, option)* to check whether the user has chosen a specific option from a selection (radio button selection, check box selection or a drop-down list). For the first parameter, specify the name of the selection item; for the second parameter, specify the option value (not its label).

```
isSelected(newsletter, "yes");
```

Access to form values is provided using variables that are made available to you automatically. If you have defined an input field with the name *email*, for example, you can access the value of this field directly via the *email* variable. You can access the current page number value with the *pageId* variable.

If, for example, you want to ensure that the user can only enter a value for the *postcode* field if the user has also entered something into the *town* field, then you can set this up by using the following function:

```
function validate () {
    if (town != "" && postcode == "") {
        return "Please also enter a postcode.";
    } else {
        return "";
    }
}
```

The following example shows you a validation script that can be used to ensure that only people aged 16 years or older can subscribe to a newsletter. Younger people can only submit the form without subscribing to the newsletter.

The corresponding form contains an input field *birthday* for the date of birth, plus a drop-down list *newsletter* with the option *yes* for subscribing to newsletter mailings.

```
function validate () {
    var age = parseAge("dd/MM/yyyy", birthday);

    if (isSelected(newsletter, "yes") && age < 16) {
        return "You must be at least 16 years old to subscribe
            to the newsletter.";
    } else {
        return "";
    }
}
```

The validation script is executed as soon as the user moves to a different form page or submits the form.

Double opt-in tab

On the *Double opt-in* tab, you can activate and configure email confirmation (also known as “double opt-in”) for end users. This functionality lets you verify that the user has access to the email address that was entered into the form. To enable this verification, an additional step is added to the form in which the user is sent a confirmation link. This link must be accessed in order to complete form entry successfully. This also means that the actions you have configured are not executed until the user has accessed the link that was sent in the email.

Check the box on this tab to activate email confirmation functionality for this form.

Properties	Validation	Double Opt-in <input checked="" type="checkbox"/>
------------	------------	---------------------------------------------------



All of the entries in the following fields will now also be validated. This validation does not take place if you unselect the check box. You can fill out the fields out anyway, however, and save the values for later.

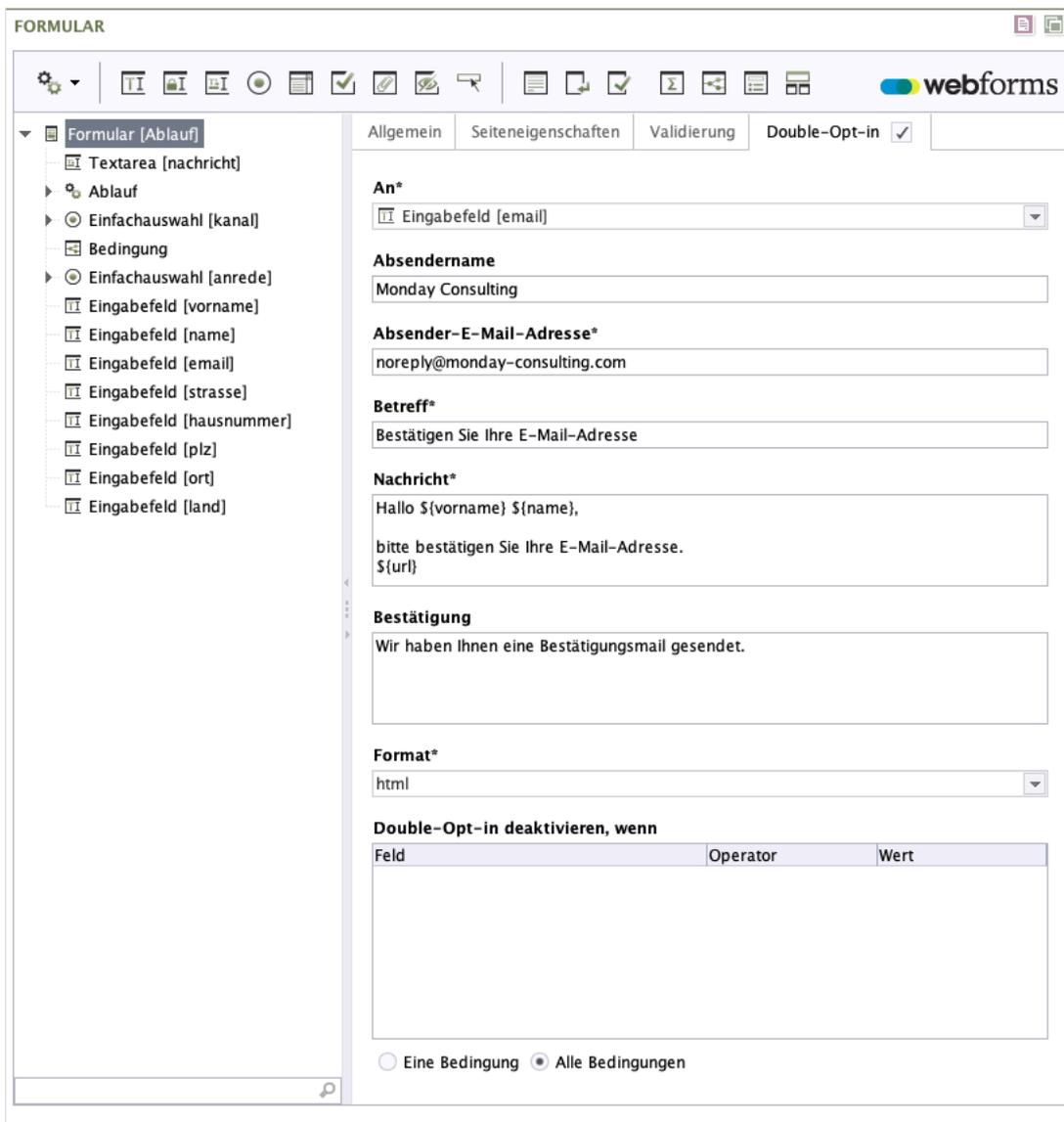


Figure 3.7. Form element – Double opt-in tab

To: This drop-down list shows you all of the form’s input fields. Pick an input field that is marked as a mandatory field and which is checked by the email validator.

Sender Name: Enter the name here that you want to use as the sender of the confirmation mail.

Sender Email Address: Enter the email address here that you want to use as the sender of the confirmation mail. Note that the email address must be valid, otherwise an error will be generated.

Subject: This is the subject that is displayed to users when they receive the mail.

Email body: This field is used to store the body of the email. To finish completing the form, the user must click on the confirmation link. You can include this link anywhere in the email body by using the variable ``${url}``. If you do not use the variable, the link is appended automatically to the end of the email body you have entered. Regardless of the *Format* selected, you can always format your mail body with FreeMarker.

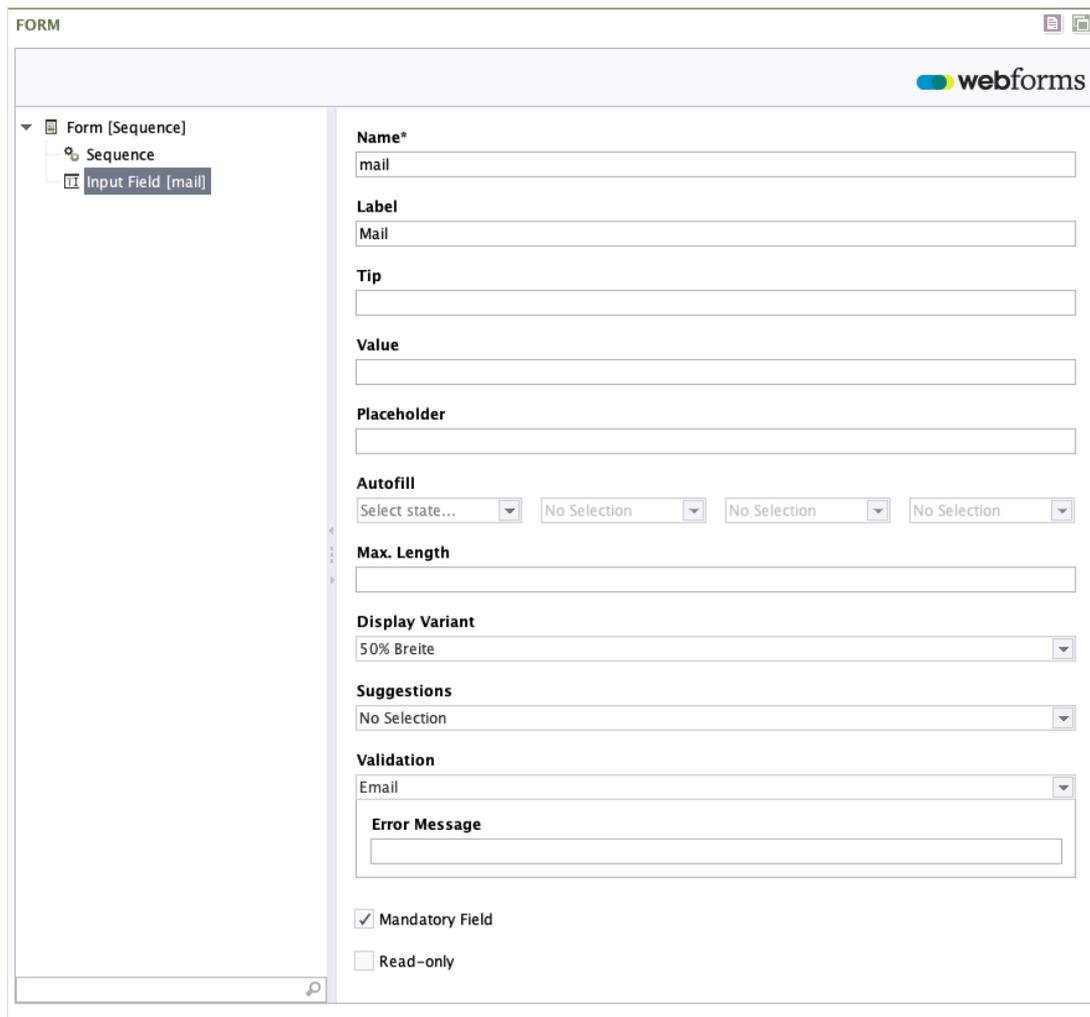
Confirmation text: When you activate email confirmation, the user will be shown a new intermediate page when the form has been filled out in full. The user must now respond to the email he or she has been sent. This field is used to store the text that will be shown on the intermediate page.

Format: This field lets you specify whether the email should be sent in HTML format or as a plain text message.

Disable Double Opt-in when:: Here you can specify the user input that causes data to be processed without first requiring confirmation by the user. The condition can be created as described in Section 3.2.11, “Condition” .

3.2.2. Input Field

The following section describes a single-line text input field.



The screenshot shows the 'webforms' configuration interface for an 'Input Field [mail]'. On the left, a tree view shows the hierarchy: 'Form [Sequence]' > 'Sequence' > 'Input Field [mail]'. The main configuration area on the right includes the following fields and options:

- Name***: Text input field containing 'mail'.
- Label**: Text input field containing 'Mail'.
- Tip**: Empty text input field.
- Value**: Empty text input field.
- Placeholder**: Empty text input field.
- Autofill**: Four dropdown menus, each set to 'No Selection'.
- Max. Length**: Empty text input field.
- Display Variant**: Dropdown menu set to '50% Breite'.
- Suggestions**: Dropdown menu set to 'No Selection'.
- Validation**: Dropdown menu set to 'Email'.
- Error Message**: Empty text input field.
- Mandatory Field**
- Read-only**

Figure 3.8. Input Field

Name: Enter the technical name of the Input field here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the Input field in the frontend.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the Input Field in the frontend.

Default Value: In this field, you can enter a piece of text to use as the default value for the Input field when the user accesses the form for the first time. The text can also contain variables (see Section 3.3, “Variables”).

Placeholder: In this field, you can enter a piece of placeholder text; this is displayed in the input field but disappears when the user clicks into the field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps users to fill out their forms faster.

If an input field is asking for a street name for an invoice address, for example, you could specify the following:



A row of four form fields. The first is a dropdown menu with 'On' selected. The second is a dropdown menu with 'Billing' selected. The third is a dropdown menu with 'Select contact type...' selected. The fourth is a dropdown menu with 'Address: Line 1' selected.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then pick the “Off” option from the first drop-down list.

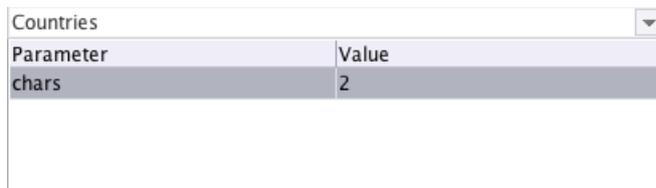


As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete>.

Max. Length: In this field, you specify the maximum number of characters that the user may enter into the Input field.

Display Variant: Use this to pick an alternative presentation style for the Input field.

Suggestions: In this field, you can specify which autocomplete entries (if any) should be displayed to users while they are typing.



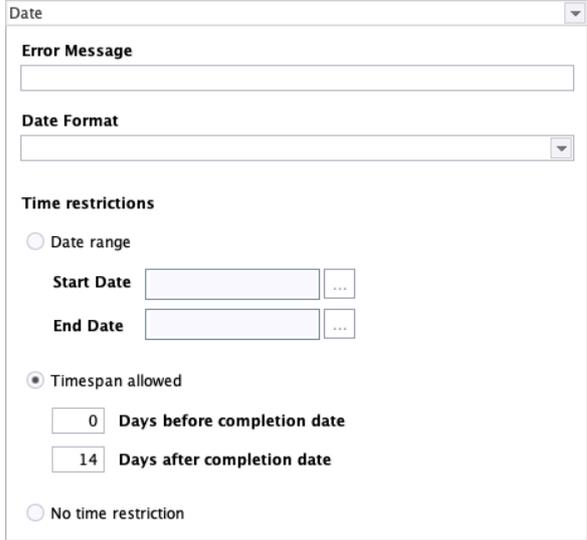
Parameter	Value
chars	2

The table gives you the option of entering additional parameters to pass to the service that creates the autocomplete list. Permissible parameters will depend on the service selected.

For the service for German and English country names contained in the standard scope of delivery, the parameter *chars* can be used to specify the minimum number of characters the user must enter into the field before an autocomplete entry is shown.

Validation: In this drop-down, you specify in which format data entry must be made, and/or specify the acceptable date and number ranges for user input. As standard, available valida-

tors include *Email*, *Date*, *No. of Characters*, *Number*, *Postcode*, *Phone Number*, *IBAN*, *BIC* and *Same Value*. In addition, you also have the option of using a regular expression to check user input. This is useful for checking data such as customer numbers, etc.

<p>Email</p>	<p>By selecting this validator, you can ensure that the value input represents a valid email address.</p>  <p>Error Message: An alternative error message that is output instead of the standard error message.</p>
<p>Date</p>	<p>This validator checks to see if the value input is a date and whether it lies in the permissible date range.</p>  <p>Error message An alternative error message that is output instead of the standard error message.</p> <p>Date format: The format in which the user must enter the date.</p> <p>The following placeholders can be used in the format string:</p> <ul style="list-style-type: none"> y Year MM Month of the year with leading zero d Day of the month H Hour of the day (0–23) m Minute of the hour s Second of the minute <p>The drop-down list already contains a list of commonly-used data formats. You can either use these directly or modify them to suit your personal requirements.</p>

The default setting is to use the format *dd/MM/yyyy*.

Date Range: Activate this setting if you want to restrict the date to a fixed date range.

Please note: If you do not select the corresponding date but enter it manually using the keyboard, you must confirm your input with the "Enter" key.

Start date: The lower date limit for the date entered by the user.

End date: The upper date limit for the date entered by the user.

Timespan allowed: Activate this setting if you want to restrict the date to a timespan relative to the form completion date.

Days before completion date: Enter the earliest date before the form completion date that can be entered by the user, using days before completion date as the unit.

Days after completion date: Enter the latest date after the form completion date that can be entered by the user, using days after completion date as the unit.

To for example allow only dates in the past, you can enter either 0 (days in the future including the date of completion) or -1 (days in the future excluding the date of completion) for the value *Days before completion date*.

No time restriction: Activate this setting if the user can enter any date without it needing to be checked.

Number of Characters

This validator checks the number of characters entered.



Number of Characters

Error Message

Min. Characters

1

Max. Characters

10

Error Message: An alternative error message that is output instead of the standard error message.

Min. Characters: The minimum number of characters that the user must enter into the field.

Max. Characters: The maximum number of characters that the user may enter into the field.

Number

This validator checks to see if the value input is a number and whether it lies in the acceptable number range.

Error Message: An alternative error message that is output instead of the standard error message.

Min: The smallest acceptable number that the user may enter.

Max: The largest acceptable number that the user may enter.

Integers only: If you activate this check box, then the user may not enter any number with decimal places.

Postcode

This validator checks to see if the user input is a valid postcode.

Error Message: An alternative error message that is output instead of the standard error message.

Country: Choice of country for which the postcode should be validated.

Country Selection: In this field, you can select a drop-down list from the form (see Section 3.2.7, “List of Choices”) with country names. When validating the postcode, the postcode format is then used for the country that has been selected by the user. If both a country and a country selection have been made, then the country selection has priority.

The postcode validator supports the formats used in the countries listed below. Note that the option value assigned to the country in the drop-down list must be its ISO 3166 country code.

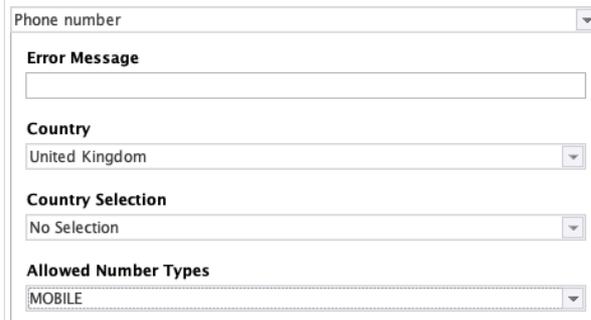
Albania (AL), Austria (AT), Belgium (BE), Bosnia and Herzegovina (BA), Bulgaria (BG), Croatia (HR), Cyprus (CY), Czech Republic (CZ), Denmark (DK), Estonia (EE), Finland (FI), France (FR), French Guiana (GF), Germany (DE), Greece (GR), Guadeloupe (GP), Hungary (HU), Iceland (IS), Ireland (IE), Italy (IT), Kosovo (RS-KM), Latvia (LV), Liechtenstein (LI), Lithuania (LT), Luxembourg (LU),

Macedonia (MK), Malta (MT), Martinique (MQ), Moldavia (MD), Montenegro (ME), Netherlands (NL), Norway (NO), Poland (PL), Portugal (PT), Réunion (RE), Romania (RO), Serbia (RS), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), Switzerland (CH), Tunisia (TN), Turkey (TR), Ukraine (UA), United Kingdom (GB)

If the country code is not listed, then the postcode is not validated.

Phone Number

This validator checks to see if the user input is a valid telephone number, and whether this number is valid for the selected country and type.



Error message: An alternative error message that is output instead of the standard error message.

Country: Choice of country for which the phone number should be validated.

Country selection: In this field, you can select a drop-down list from the form (see Section 3.2.7, “List of Choices”) with country names. When validating the phone number, a final check is made to see if the number is a valid phone number from the selected country. If both a country and a country selection have been made, then the country selection has priority.

This validator supports formats from nearly every country in the world. Note that the option value assigned to the country in the drop-down list must be its ISO 3166 country code.

Validation will fail for country codes that are not listed.

Permitted number types: From this list, you can select the permissible number types that the phone number can have.

Formats from the number types listed below are supported.

Fixed line (FIXED_LINE), mobile (MOBILE), fixed line or mobile (FIXED_LINE_OR_MOBILE), toll-free (TOLL_FREE), premium rate (PREMIUM_RATE), shared cost (SHARED_COST), VOIP (VOIP), personal number (PERSONAL_NUMBER), pager (PAGER), universal access numbers (UAN), voicemail (VOICEMAIL), unknown (UNKNOWN)



For some countries, the FIXED_LINE and MOBILE types may be ambiguous. In these cases, you may also need to select the FIXED_LINE_OR_MOBILE type for validation to execute correctly. Note that the FIXED_LINE_OR_MOBILE type is *not* a combination of FIXED_LINE and MOBILE but a separate type.

If the number type is not listed, then the phone number is not validated.

IBAN

This validator checks to see if the user has input a valid International Bank Account Number (IBAN).

IBAN

Error Message

Country

United Kingdom

Country Selection

No Selection

Error Message: An alternative error message that is output instead of the standard error message.

Country: When a country is specified, an additional check is made to see whether the IBAN format matches the format typically used in the selected country. If no country is specified, then a check is only made to verify a match with the basic IBAN format.

Country Selection: In this field, you can select a drop-down list from the form (see Section 3.2.7, “List of Choices”) containing country names. The validation of the IBAN is then based on the country selected by the user. If both a country and a country selection are selected, the country selection has priority.

BIC

This validator checks to see if the user has input a valid international Bank Identifier Code (BIC).

BIC

Error Message

Error Message: An alternative error message that is output instead of the standard error message.

Regular
Expression

This validator uses a regular expression to check user input.

A regular expression (which can also be abbreviated as regexp or regex) is a character string that uses syntactical rules to define character string entities.

Error Message: An alternative error message that is output instead of the standard error message.

Expression: In this field, you can enter or select a regular expression that defines the input format you require. The drop-down list already contains regular expressions suitable for a range of common applications. You can either use these directly or modify them to suit your personal requirements.

You will find a short guide to the authoring of regular expressions at the following URL: https://en.wikipedia.org/wiki/Regular_expression

Same Value

This validator checks whether the same value has been entered into two Input fields.

Error Message: An alternative error message that is output instead of the standard error message.

Compare with: In this field, you can select a second Input field on the form. During validation, the system checks whether the same value has been entered for both fields.

European number VAT

This validator checks to see if the user input is a valid European VAT number.

Error message: An alternative error message that is output instead of the standard error message.

Country: Choice of country for which the VAT number should be validated.

Country selection: In this field, you can select a drop-down list from the form (see Section 3.2.7, “List of Choices”) with country names. When validating the VAT number, the VAT number format is then used for the country that has been selected by the user. If both a country and a country selection have been made, then the country selection has priority.

The VAT number validator supports the formats used in the countries listed below. Note that the option value assigned to the country in the drop-down list must be its ISO 3166 country code.

Austria (AT), Belgium (BE), Bulgaria (BG), Croatia (HR), Cyprus (CY), Czech Republic (CZ), Denmark (DK), Estonia (EE), Finland (FI), France (FR), Germany (DE), Greece (EL), Greece (GR), Hungary (HU), Ireland (IE), Italy (IT), Latvia (LV), Lithuania (LT), Luxembourg (LU), Malta (MT), Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovak Republic (SK), Slovenia (SI), Spain (ES), Sweden (SE)

You can also use this validator to check the VAT numbers of the following non-EU countries:

Serbia (RS), Russia (RU), Norway (NO), United Kingdom of Great Britain and Ireland (GB), Switzerland (CH)

Mandatory Field: If you activate this check box, then the user will be required to enter at least a single character into the Input field. This character cannot be a space character, however. The user will be unable to submit the form or continue on to the form’s next page until the field has been filled-out.

Read-only: If you activate this check box, then the user will be prevented from changing the text in the field. This is often used if the Input field has been pre-assigned a variable (see Section 3.3, “Variables”) as a default value.

3.2.3. Password field

The Password field is a single-line input field for passwords, in which the numbers and letters entered by the user are shown as solid dots.

The screenshot shows the 'FORMULAR' configuration window for a 'Password' field. The left sidebar shows a tree view with 'Form [Sequence]' expanded to show 'Sequence', 'Password [password]', and 'Password [passwort_wdh]'. The main configuration area includes the following sections:

- Name***: A text input field containing 'password'.
- Label**: A text input field containing 'Password'.
- Tip**: An empty text input field.
- Autofill**: A section with four dropdown menus: 'On', 'Select mode...', 'Select contact ty...', and 'New password'.
- Placeholder**: An empty text input field.
- Display Variant**: A dropdown menu set to 'No Selection'.
- Validation**: A dropdown menu set to 'Password'.
- Error Message**: An empty text input field.
- Compare to**: A dropdown menu set to 'Password [passwort_wdh]'.
- Minimum Length**: An empty text input field.
- Special Characters**:
- Numbers**:
- Uppercase and Lowercase**:
- Mandatory Field**:

Figure 3.9. Password

Name: Enter the technical name of the Password field here.

Label: In this field, you can enter a piece of label text that will be displayed above the Password field in the frontend.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the Password field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like passwords, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the type of password expected for a specific field.

If an input field is asking for a new password, for example, you could specify the following:

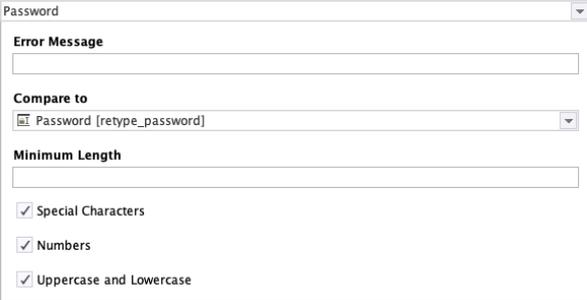
The screenshot shows the 'Autofill' configuration section with four dropdown menus: 'On', 'Select mode...', 'Select contact type...', and 'New password'.

If you want to stop the browser saving the password automatically, then pick the “Off” option from the first drop-down list.

Placeholder: In this field, you can enter a piece of placeholder text; this is displayed in the Password field but disappears when the user clicks into the field.

Display Variant: Use this to pick an alternative presentation style for the Password field.

Validation: If you have a scenario where you need to specify security requirements for the password chosen by the user, this drop-down list lets you select a password validator.

Password	<p>This validator checks whether the password entered complies with the security requirements.</p>  <p>Error Message: An alternative error message that is output instead of the standard error message.</p> <p>Minimum Length: The minimum number of characters that the password must contain.</p> <p>Compare to: In this field, you can select a second Password field on the form. During validation, the system checks whether the same value has been entered for both fields.</p> <p>Special Characters: If you activate this check box, then the user must enter a password containing a special character.</p> <p>Numbers: If you activate this check box, then the user must enter a password containing at least one number.</p> <p>Lowercase and Uppercase: If you activate this check box, then the user must enter a password containing lowercase and uppercase letters.</p>
----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Mandatory Field: If you activate this check box, then the user will be required to enter at least a single character into the Password field. This character cannot be a space character, however. The user will be unable to submit the form or continue on to the form's next page until the field has been filled-out.

3.2.4. Textarea

A Textarea is a multi-line text input field, which can contain line breaks.

The screenshot shows the 'FORMULAR' configuration window for a 'Text Area [nachricht]' field. The configuration options are as follows:

- Name***: Input field containing 'nachricht'.
- Label**: Input field containing 'Ihre Fragen an uns'.
- Tip**: Empty input field.
- Value**: Large empty text area for default value.
- Autofill**: Four dropdown menus, all set to 'No Selection'.
- Placeholder**: Empty input field.
- Max. Length**: Empty input field.
- Rows**: Input field containing '6'.
- Cols**: Empty input field.
- Display Variant**: Dropdown menu set to 'No Selection'.
- Mandatory Field**: checkbox.
- Read-only**: checkbox.

Figure 3.10. Textarea

Name: Enter the technical name of the Textarea here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the Textarea field in the frontend.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the Textarea field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Default Value: In this field, you can enter a piece of text to use as the default value for the field when the user accesses the form for the first time. The text can also contain variables (see Section 3.3, “Variables”).

Autofill: You can use the *Autofill* function to utilise the autofill functionality offered by the browser. Browsers save form data entered previously like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field.

Placeholder: In this field, you can enter a piece of placeholder text; this is displayed in the input field but disappears when the user clicks into the field.

Max. Length: In this field, you specify the maximum number of characters that the user may enter into the Textarea.

Rows: The number of visible rows.

Cols: The number of visible columns.

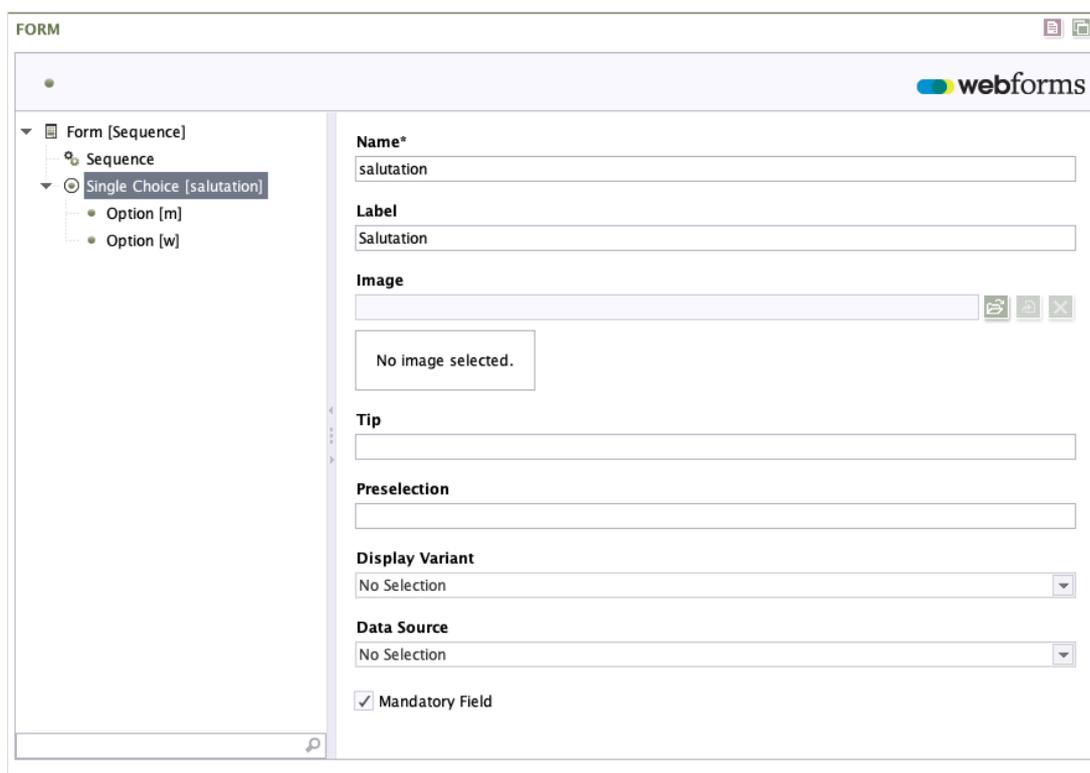
Display Variant: Use this to pick an alternative presentation style for the Summary.

Mandatory Field: If you activate this check box, then the user will be required to enter at least a single character into the Textarea. This character cannot be a space character, however. The user will be unable to submit the form or continue on to the form's next page until the field has been filled-out.

Read-only: If you activate this check box, then the user will be prevented from changing the text in the field.

3.2.5. Single Choice

A type of select field where the user can choose one of several options (see Section 3.2.8, "Option"). Each option is shown as a radio button.



The screenshot shows the 'webforms' configuration interface. On the left, a tree view shows the hierarchy: Form [Sequence] > Sequence > Single Choice [salutation] > Option [m] and Option [w]. The 'Single Choice [salutation]' field is selected. The main configuration area on the right includes the following fields:

- Name***: A text input field containing 'salutation'.
- Label**: A text input field containing 'Salutation'.
- Image**: A text input field with a file selection icon and a 'No image selected.' message below it.
- Tip**: A text input field.
- Preselection**: A text input field.
- Display Variant**: A dropdown menu with 'No Selection' selected.
- Data Source**: A dropdown menu with 'No Selection' selected.
- Mandatory Field**: A checked checkbox.

Figure 3.11. Single Choice

Name: Enter the technical name of the select field here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the select field in the frontend.

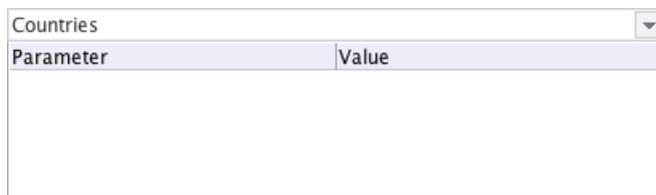
Image: You have the option of storing an image here that will be shown together with the select field.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the select field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Preselection: In this field, you can enter the value of an option that should be preselected the first time that the form is accessed. This function enables the creation of a Single Choice field whose options are determined from a data source (see below) at runtime.

Display Variant: Use this to pick an alternative presentation style for the select element.

Data Source: In this field, you can select a data source from which the options in the select field are determined dynamically at runtime.



The screenshot shows a configuration window for a select field. At the top, there is a dropdown menu labeled 'Countries'. Below it is a table with two columns: 'Parameter' and 'Value'. The table is currently empty.

Parameter	Value
-----------	-------

The table gives you the option of entering additional parameters to pass to the data source. Permissible parameters depend on the data source selected. No parameters are required by the data sources supplied in the standard scope of delivery.

Mandatory Field: If you activate this check box, then the user will be required to select an option. Users will be unable to submit the form or continue on to the form’s next page until they have selected an option.

3.2.6. Multiple Choices

This is a type of select field where the user can make multiple selections (see Section 3.2.8, “Option”) from a predefined set of options. Each option is shown as a check box.

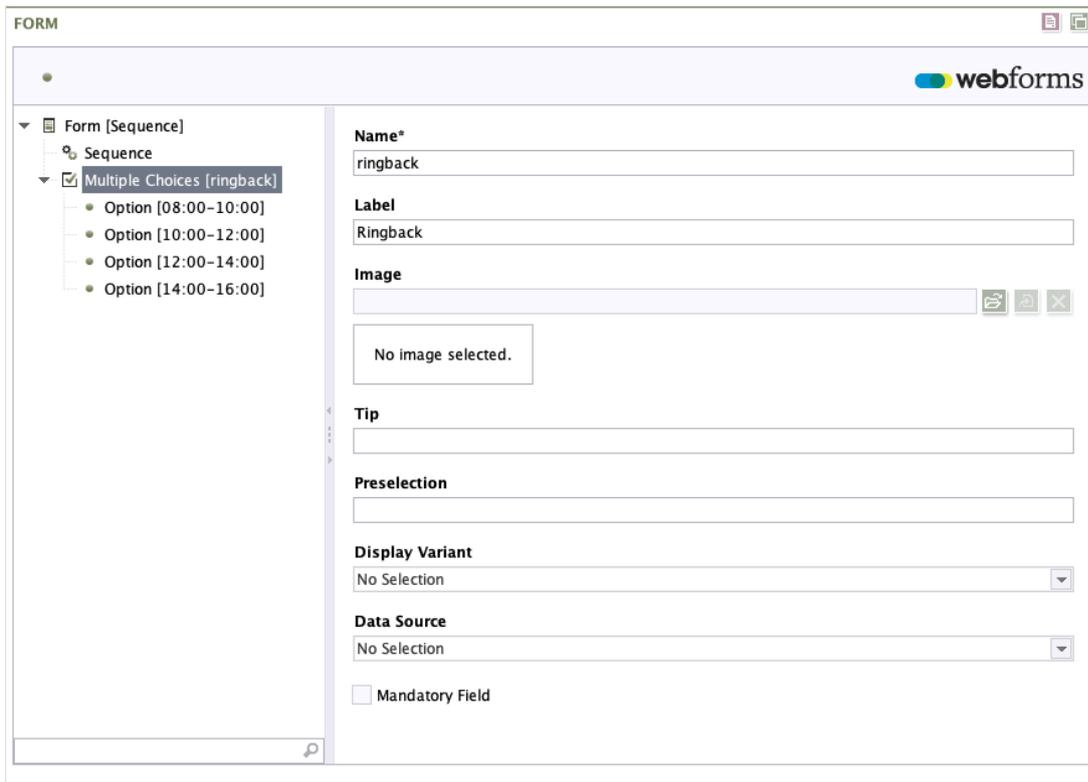


Figure 3.12. Multiple Choices

Name: Enter the technical name of the select field here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the select field in the frontend.

Image: You have the option of storing an image here that will be shown together with the select field.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the select field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Preselection: In this field, you can enter the value of one or more options that should be preselected the first time that the form is accessed. This function enables the creation of a Multiple Choices field whose options are determined from a data source (see below) at runtime.

Display Variant: Use this to pick an alternative presentation style for the select element.

Data Source: In this field, you can select a data source from which the options in the select field are determined dynamically at runtime.

Countries	
Parameter	Value

The table gives you the option of entering additional parameters to pass to the data source. Permissible parameters depend on the data source selected. No parameters are required by the data sources supplied in the standard scope of delivery.

Mandatory Field: If you activate this check box, then the user will be required to select at least one option. Users will be unable to submit the form or continue on to the form's next page until they have selected an option.

3.2.7. List of Choices

A type of select field where the user can make one or more choices (see Section 3.2.8, "Option") from a predefined set of options. The individual options are shown as a drop-down list.

The screenshot shows the 'FORMULAR' configuration window for a 'List of Choices' field. The left sidebar shows a tree view with 'Form [Sequence]' expanded to 'List of Choices [thema]', which contains four options: 'Option [EMPTY_VALUE]', 'Option [vorschlag]', 'Option [kritik]', and 'Option [frage]'. The main configuration area on the right includes the following fields and controls:

- Name***: Text input field containing 'thema'.
- Label**: Text input field containing 'Thema'.
- Image**: Image selection area with a 'No image selected.' message and icons for file selection, refresh, and close.
- Tip**: Text input field.
- Preselection**: Text input field.
- Autofill**: Four dropdown menus, each currently set to 'No Selection'.
- Display Variant**: Dropdown menu set to 'No Selection'.
- Data Source**: Dropdown menu set to 'No Selection'.
- Mandatory Field**: Checked checkbox.
- Multiple Choices**: Unchecked checkbox.

Figure 3.13. List of Choices

Name: Enter the technical name of the select field here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the select field in the frontend.

Image: You have the option of storing an image here that will be shown together with the select field.

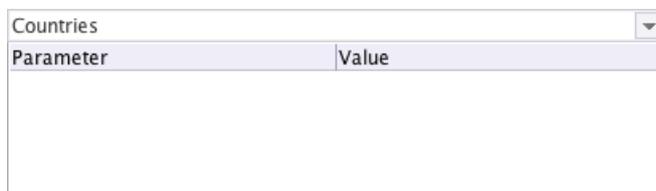
Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the select field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, "Markdown").

Preselection: In this field, you can enter the value of one or more options that should be preselected the first time that the form is accessed. This function enables the creation of a drop-down list whose options are determined from a data source (see below) at runtime. If, for example, you want to preselect the countries *Germany* and *France* in a dynamically-filled drop-down list of countries, then the values *DE,FR* need to be entered here.

Autofill: You can use the *Autofill* function to utilise the autofill functionality offered by the browser. Browsers save form data entered previously like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field.

Display Variant: Use this to pick an alternative presentation style for the Drop-Down List.

Data Source: In this field, you can select a data source from which the options in the select field are determined dynamically at runtime.



The screenshot shows a form field with a dropdown menu labeled 'Countries'. Below the dropdown is a table with two columns: 'Parameter' and 'Value'. The table is currently empty.

Parameter	Value
-----------	-------

The table gives you the option of entering additional parameters to pass to the data source. Permissible parameters depend on the data source selected. No parameters are required by the data sources supplied in the standard scope of delivery.

The data sources here are not FirstSpirit data sources but web services provided by Monday Webforms.

The options for selection are read from the data source when the user accesses the form. In the default setup, Webforms includes a data source for German and English country names. All of the countries supplied by this data source can be used in conjunction with the input field's postcode validator (see Section 3.2.2, "Input Field"). The data sources available are specified in the form's section template.

Multiple Choices: If you activate this check box, then the user will be able to choose multiple options from the list.

Mandatory Field: If you activate this check box, then the user will be required to select at least one option. Users will be unable to submit the form or continue on to the form's next page until they have selected an option.

3.2.8. Option

This element represents an entry that can be selected by the user from a List of Choices, a Single Choice or a Multiple Choices select field.

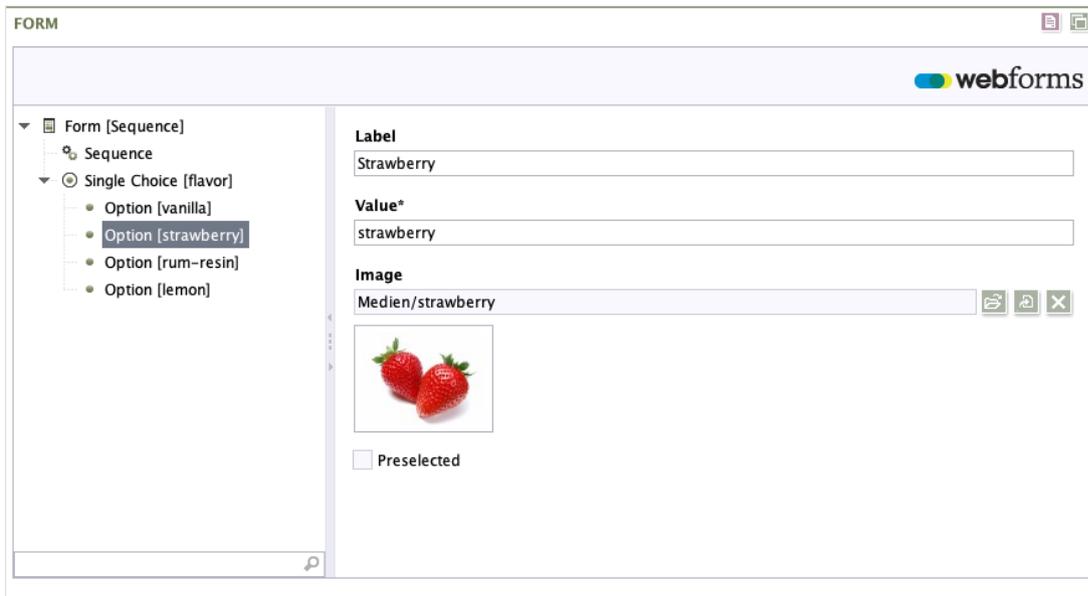


Figure 3.14. Option

Label: The label that is displayed to the user.

Value: The value that is submitted if this option has been selected. This value does not need to be the same as the label.

If you specify “EMPTY_VALUE” as the value here, then this option will be ignored in later processing steps, even if selected by the end user. This function can be used to insert a “Please select” option, for example.

Image: You have the option of storing an image here that will be shown together with the option.

Preselected: If you activate this check box, then the option will be preselected.

3.2.9. Hidden Field

If you use a Hidden field, you enter information into the form that is not displayed the user. When the form is submitted, this information is submitted along with the data from the other fields.

If a hidden field is assigned the variable `serverDate`, for example, this enables you to determine when the person accessed the form before submitting it.

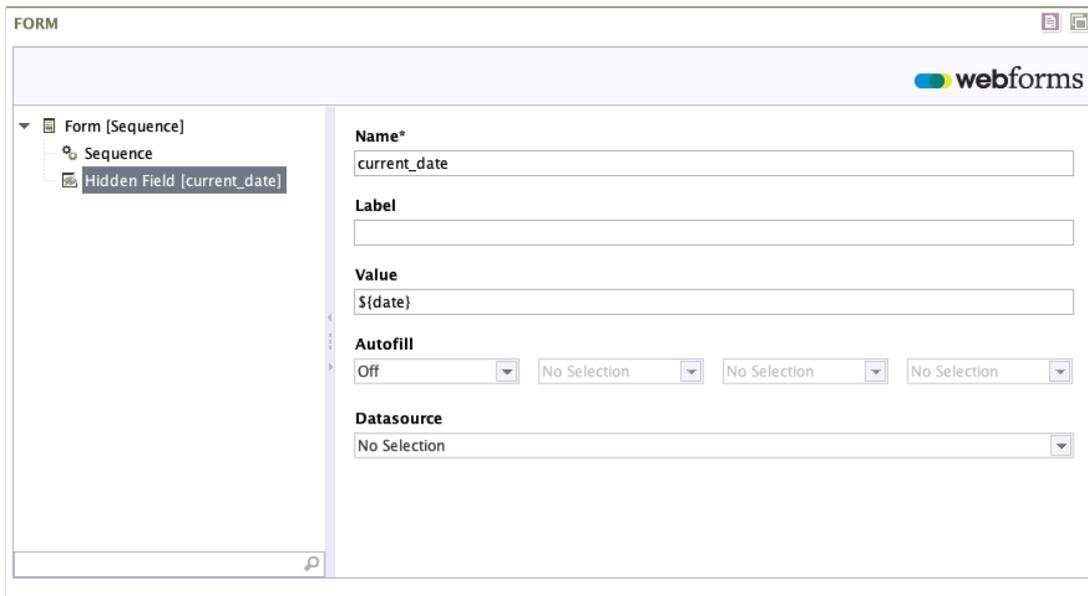


Figure 3.15. Hidden Field

Name: Enter the technical name of the Hidden field here.

Label: In this field, you can enter a piece of label text; this is then displayed in the Summary, for example.

Value: Enter the content for the Hidden field into this field. This content can also include one or more variables (see Section 3.3, “Variables”).

Autofill: You can use the *Autofill* function to utilise the autofill functionality offered by the browser. Browsers save form data entered previously like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field.

Data source: In this field, you can select a data source that determines the value of the field dynamically at runtime.

Countries	
Parameter	Value

The table gives you the option of entering additional parameters to pass to the data source. Permissible parameters depend on the data source selected. No parameters are required by the data sources supplied in the standard scope of delivery.

The data sources here are not FirstSpirit data sources but web services provided by Monday Webforms.

3.2.10. Upload field

This element lets the user upload a file to the form. Depending on the action selected, uploaded files can be sent as an email attachment or stored in the web server's file system, for example.

The screenshot shows the 'FORM' editor interface for configuring an 'Upload File' field. The left-hand pane displays a hierarchical tree view with 'Form [Sequence]' expanded to show 'Sequence' and 'File [file]'. The right-hand pane is the configuration area, featuring the 'webforms' logo at the top right. The configuration fields include: 'Name*' (text input: file), 'Label' (text input: File), 'Tip' (empty text input), 'Display Variant' (dropdown menu: No Selection), 'Validation' (dropdown menu: File), 'Error Message' (empty text input), 'Allowed File Types' (dropdown menu: doc), 'Max. File Size (KB)' (text input: 500), and 'Max. Number of Files' (text input: 5). At the bottom, there are three checked checkboxes: 'Mandatory Field', 'Multiple Files', and 'Upload files automatically'.

Figure 3.16. Upload File

Name: Enter the technical name of the Upload File field here.

Label: In this field, you can enter a piece of label text; this is then displayed in front of or above the Upload File field in the frontend.

Tip: In this field, you can enter an optional piece of note text, which describes how this form field is to be completed. This note is displayed below the field in the frontend. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Display Variant: Use this to pick an alternative presentation style for the Upload File field.

Validation: If you want to permit only certain file types or file sizes, then this drop-down list gives you the opportunity of selecting a file validator.

File	This validator checks the size and the type of an uploaded file. Note that the validator does not determine the file type by the file extension
------	-------------------------------------------------------------------------------------------------------------------------------------------------

but by analysing specific sequences in the file content. This approach ensures that the file type check cannot be bypassed by simply renaming the file.



The screenshot shows a configuration form for a file upload field. It includes a dropdown menu labeled 'File', an 'Error Message' text input field, an 'Allowed File Types' dropdown menu with 'doc' selected, a 'Max. File Size (KB)' text input field with '500' entered, and a 'Max. Number of Files' text input field with '5' entered.

Error Message: An alternative error message that is output instead of the standard error message.

Max. File Size (KB): The maximum permissible file size in kilobytes (KB).

Allowed File Types: From this list, you can select the permissible file types that may be uploaded using the file field.

Max. Number of Files: Here, enter the maximum number of files that a user can upload simultaneously by using this field.

Mandatory Field: If you activate this check box, then the user will be required to upload a file. Users will not be able to submit the form or continue on to the form's next page until they have uploaded a file.

Multiple Files: Activate this checkbox if you would like to give the user the option of using this field to upload more than one file.

Upload files automatically: If you activate this check box, then files will be uploaded to the server after selection without any further action on the part of the user. If the check box is deactivated, then the user must start the file upload by clicking an additional *Upload* button.



In the browser's selection dialog box, users can then select multiple files by holding the Shift key down while selecting files.

3.2.11. Condition

This Condition element enables you to change the state of individual form elements, depending on input entered by users.

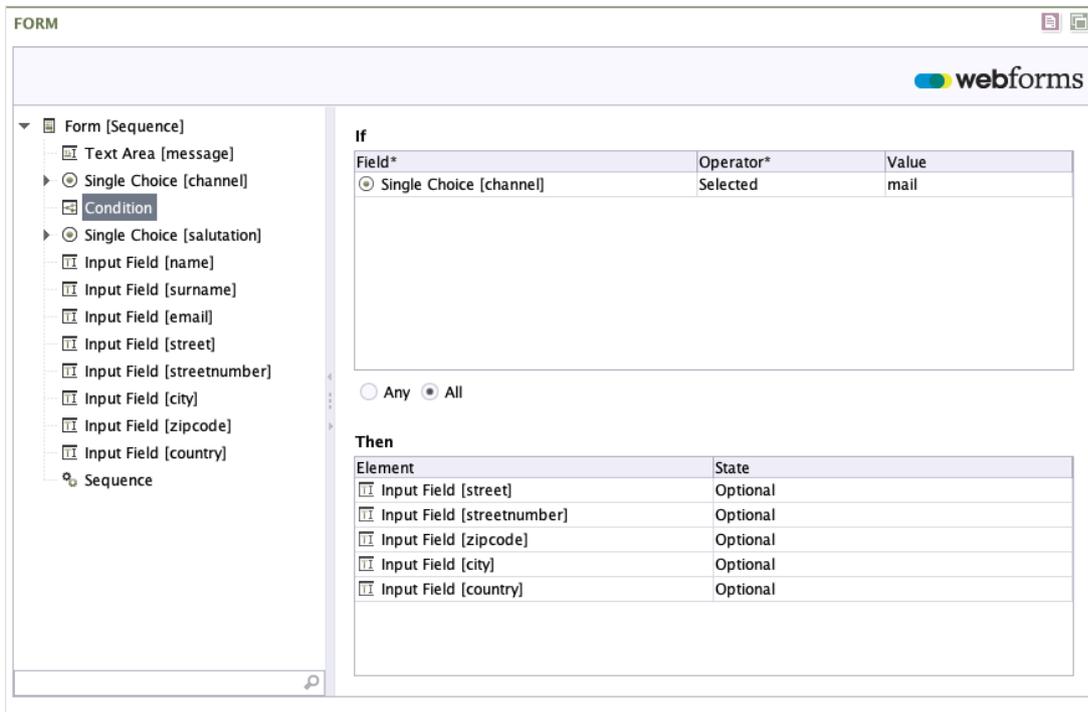


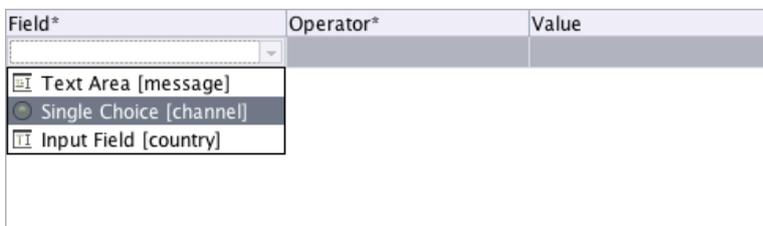
Figure 3.17. Condition

If: A condition that must be satisfied in order for the form elements listed in the *Then* field to acquire the state defined for them.

To add a new form element to the condition, right-click the table that is displayed.



If you double-click the first cell of the row that has just been added, a drop-down list is shown. From this list, select the form element that you wish to include in the condition.



You use the *Operator* column to specify the logical operator for the condition. This is then applied to the reference value specified in the *Value* column. The operators offered to you here will depend on the field content and whether the field is a select field or a text input field.

Field*	Operator*	Value
<input checked="" type="radio"/> Single Choice [channel]	<input type="text"/>	
	Not Selected Selected Is Empty Is Filled Selected more than Selected less than	

In the *Value* column, enter the reference value or, in the case of a drop-down list, select the corresponding option.

Field*	Operator*	Value
<input checked="" type="radio"/> Single Choice [channel]	Selected	<input type="text" value="mail"/>
		mail post

If more than one logical condition applies, you can use the following select field to decide whether one or all conditions must be satisfied.

Any
 All

Then: The form elements contained in this list acquire the state defined for them if the above-mentioned condition applies.

To add a new form element to the list, right-click with the mouse on the condition field.

Element	State
<input type="button" value="Add Element"/> <input type="button" value="Remove Element"/>	

If you double-click the first cell of the row that has just been added, a drop-down menu is shown. From this list, select the form element whose state you want to change, depending on the fulfilment of the previously-defined condition(s).

Element	State
<input type="text"/>	
<input checked="" type="radio"/> Text Area [message] <input checked="" type="radio"/> Single Choice [channel] <input checked="" type="radio"/> Input Field [country] <input type="checkbox"/> Submit <input type="checkbox"/> Cancel <input type="checkbox"/> Exit	

You use the *State* column to specify the state you want the element to have when the condition is fulfilled. Depending on the form element selected, several states will be offered to you here.

Element	State
 Input Field [country]	<input type="text"/>
	Visible
	Hidden
	Optional
	Mandatory
	Writable
	Read-Only
	Enabled
	Disabled

If the condition is not fulfilled, the field is automatically given the opposite state. As an example: if a field should be shown when a certain option is selected, the field is automatically hidden (hidden = opposite of shown) if this option has not been selected.



Note: You can specify the states *Optional* / *Mandatory* and *Writable* / *Read-Only* for a form element only if this element has not already been defined in the element definition as a *Mandatory Field* or as *Read-Only*.

Hidden form fields are also hidden on the Summary pages and in any emails that are sent. Values from deactivated form fields are ignored.

3.2.12. Page Break

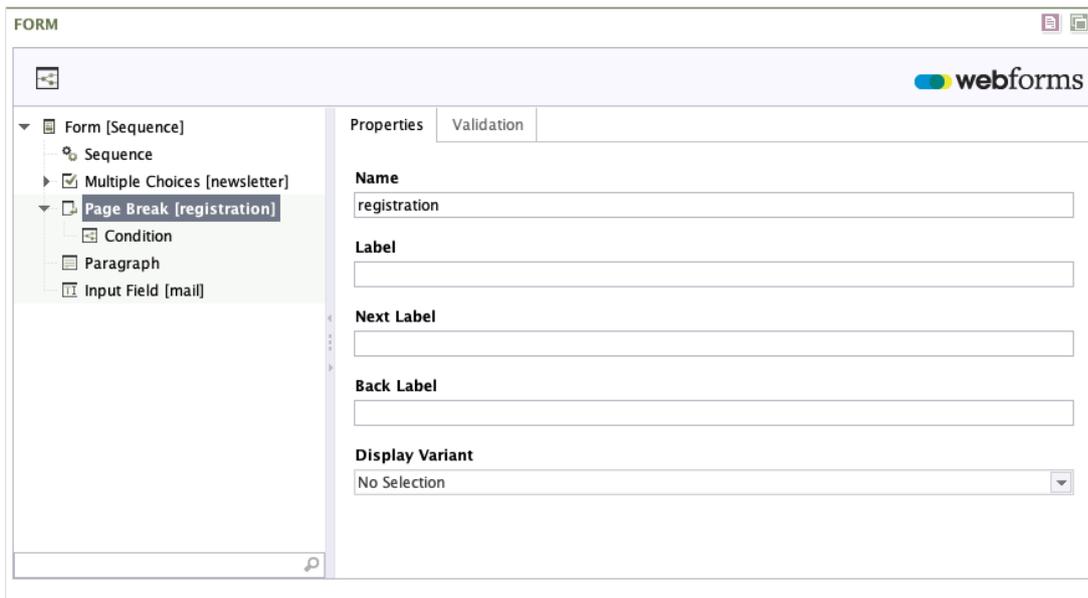
You use the *Page Break* element to construct multi-page forms. The element can be inserted at any position in the form. All of the form elements below it will be displayed on the next page. With multi-page forms, the user is shown "Next" and "Back" buttons. The user can then use these buttons to page back and forth between the individual form pages.

Forms can consist of as many pages as necessary, and you can even create empty pages.

If a page should only be displayed if the user has made specific kinds of input on the previous pages of the form, then you can define a condition to provide this functionality.

The detail view of the page break is split over the *Properties* and *Validation* tabs. The various configuration options are explained in the following sections.

Properties tab



The screenshot shows the 'FORM' editor interface with the 'Properties' tab selected. On the left, a tree view shows the form structure: Form [Sequence] > Sequence > Multiple Choices [newsletter] > Page Break [registration]. The 'Page Break [registration]' element is selected. The main area displays the following properties:

- Name:** registration
- Label:** (empty text field)
- Next Label:** (empty text field)
- Back Label:** (empty text field)
- Display Variant:** No Selection (dropdown menu)

Figure 3.18. Page break – Properties tab

Name: Enter the technical name of the form page here.

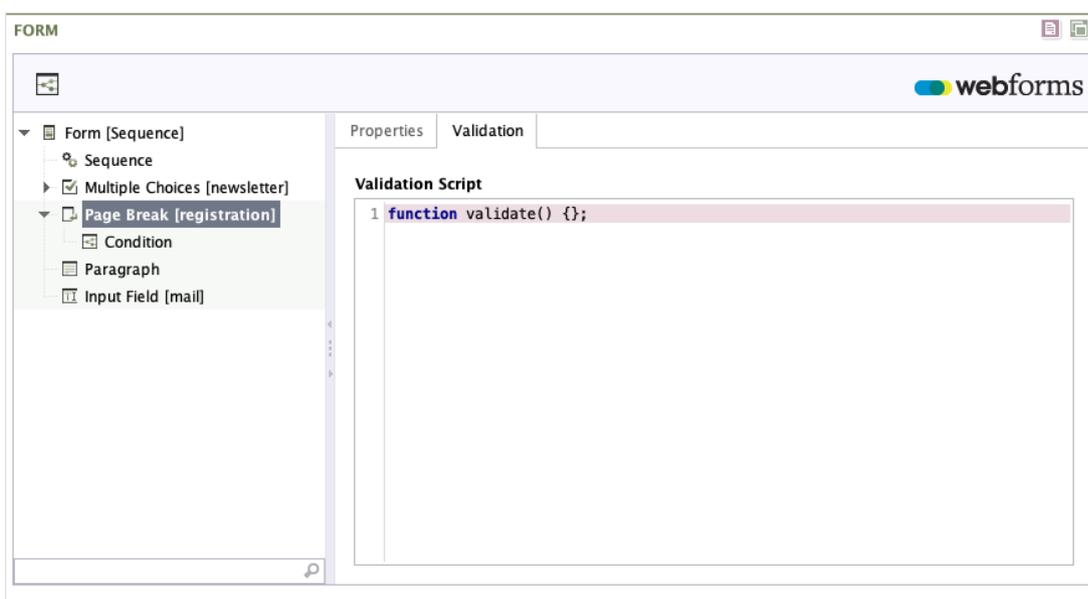
Label: This text is output as a heading shown to the user on the form page.

Next Label: This text is displayed on the “Next” button that is shown on the first page of a multi-page form. This text item is not used for forms that have only one page.

Back Label: This text is displayed on the “Back” button.

Display Variant: Use this to pick an alternative presentation style for the form page.

Validation tab



The screenshot shows the 'FORM' editor interface with the 'Validation' tab selected. The tree view on the left is the same as in Figure 3.18. The main area displays the 'Validation Script' field with the following code:

```
1 function validate() {};
```

Figure 3.19. Page break – Validation tab

Validation Script: You can use the validation script to validate the data entered into the form. Unlike field validators, which verify only the value of a single field, this function gives you the opportunity to simultaneously check the values of multiple fields and to create dependencies between them (see also Section 3.2.1, “Form”).

The validation script is executed as soon as the user moves to a different form page or submits the form.

3.2.13. Page break condition

Placing the condition element under a Page Break makes it possible for you to dynamically show or hide the form page depending on user input.

If there are multiple conditions for a form page, their evaluation occurs in the order in which they are assigned below the Page Break element. Note that individual condition elements are linked using OR. This means that evaluation stops as soon as one condition is met, and the following conditions are then no longer evaluated.

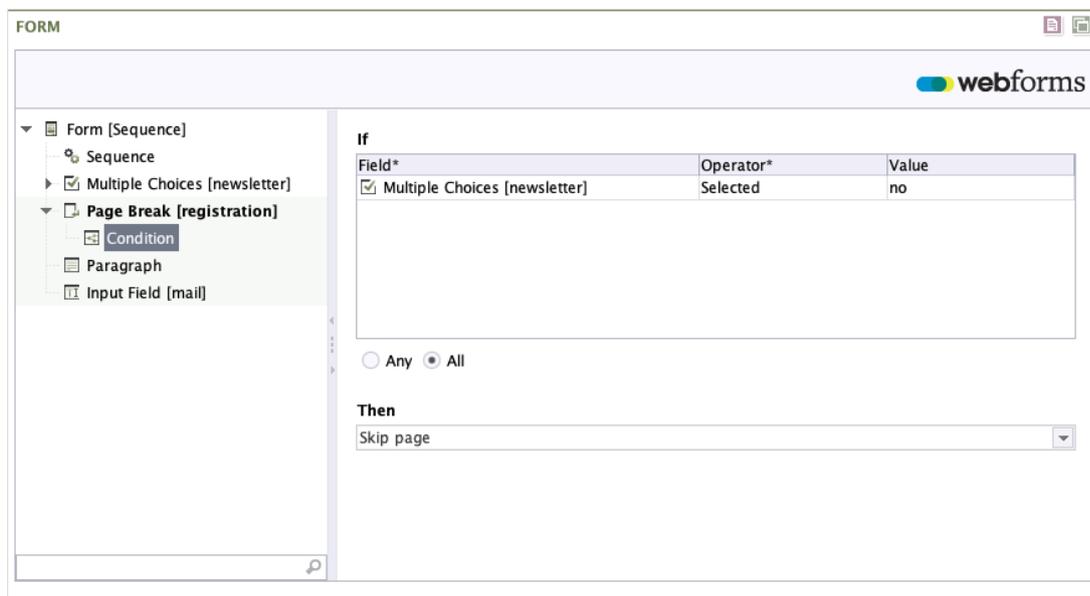


Figure 3.20. Page break condition

If: If this condition is satisfied, this page is not displayed when the user clicks the “Next” button on the previous page.

If the page hidden in this way is the last page of the form, then the form will be submitted when the user clicks the “Next” button.

The condition can be created as described in Section 3.2.11, “Condition”.



Take care to ensure that *Calculated Values* always return a value if you use them within a condition. If field input is used to calculate the value, for example, then you must ensure that the fields concerned are mandatory fields.

Then: In this drop-down list, you can specify the action the system should take if the above-mentioned condition is satisfied and the user clicks the "Next" button on the previous page.



This list only displays form pages where the field *Name* is filled out.

3.2.14. Calculated Value

This element provides a value that cannot be entered directly by the user: instead, it is calculated from other form values using code written in JavaScript.

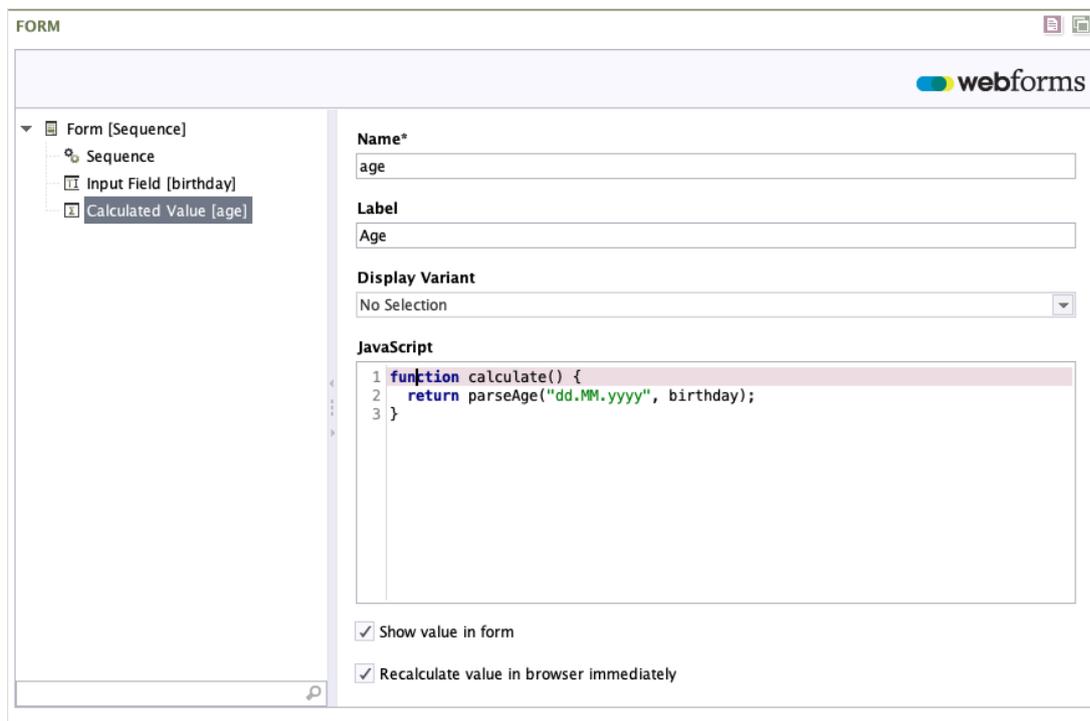


Figure 3.21. Calculated Value

Name: Enter the technical name of the Calculated Value here.

Label: In this field, you can enter a piece of label text; this can then be displayed in the Summary, for example.

Display Variant: Use this to pick an alternative presentation style for the Calculated Value field.

JavaScript: When you first create the element, the field *JavaScript* contains the empty method `calculate()`. This is a JavaScript function that is executed every time the form is submitted (page change and click on the "Submit" button). You can enter the calculation logic within the function.

If you would like to calculate the age of a person for a given date of birth – so as to then be able to use this in a condition, for example – then you can achieve this with the following

function. In your form, include an input field *birthdate*, in which users have to enter their date of birth.

```
function calculate() {  
    return parseAge("yyyy/MM/dd", birthdate);  
}
```

Show value in form: If you activate this check box, then the result of the calculation will be displayed in the form.

Recalculate value in browser immediately: Activate this check box to specify that the value will be recalculated immediately, i.e. as soon as the user has changed the input that is being used to calculate the value. In this case, the value is recalculated in the browser, i.e. the user does not need to resubmit the form.



Note that you must also activate this check box if you are using the value in a condition.

3.2.15. Paragraph

This element provides a multi-line block of text that cannot be modified by the user. This can be used to offer advice or give explanations, for example.

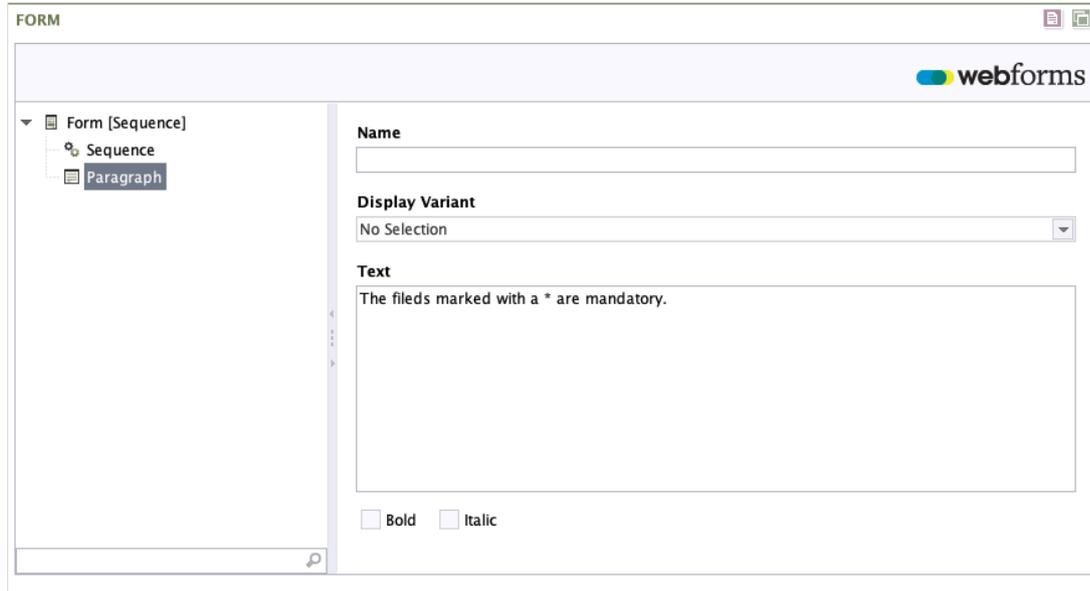


Figure 3.22. Paragraph

Name: Enter the technical name of the Paragraph field here.

Text: This text will be shown to the user in the form. The text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Display Variant: Use this to pick an alternative presentation style for the Paragraph field.

Format: The font style that should be used to format the text block.

3.2.16. Captcha

This element adds a “CAPTCHA” (acronym for “Completely Automated Public Turing test to tell Computers and Humans Apart”) to the form. Captchas are used to ensure that the form is being filled out by a human and not by an automated system.

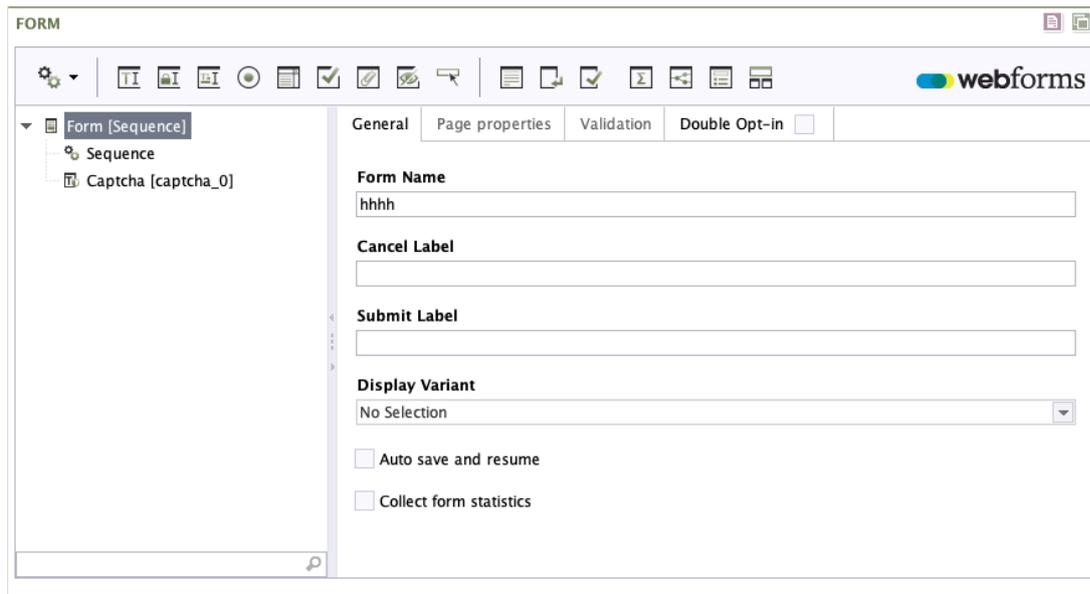


Figure 3.23. Captcha

Name: Enter the technical name of the Captcha field here.

Label: In this field, you can enter a piece of label text that will be displayed above the Captcha element in the frontend.

Tip: You can enter an optional note text here. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Error Message: An alternative error message that is output instead of the standard error message.

3.2.17. Button

Clickable button that can be used to execute any JavaScript function as desired when the user clicks the button.

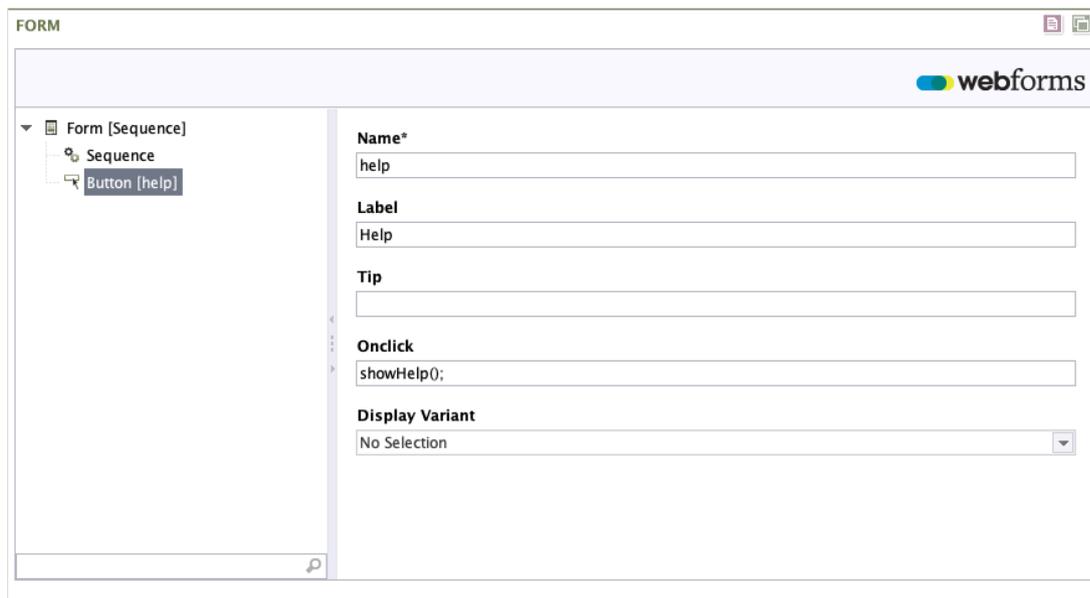


Figure 3.24. Button

Name: Enter the technical name of the Button field here. This is used only in conjunction with conditions.

Label: Enter a piece of label text into this field; this is then displayed on the Button.

Tip: You can enter an optional note text here. The note text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

OnClick: To specify what should happen when the button is clicked, use the event handler *onclick* here, so as to respond to the click with JavaScript.

Display Variant: Use this to pick an alternative presentation style for the Button field.

3.2.18. Fieldset

The Fieldset element allows you to group multiple form elements under a common heading.

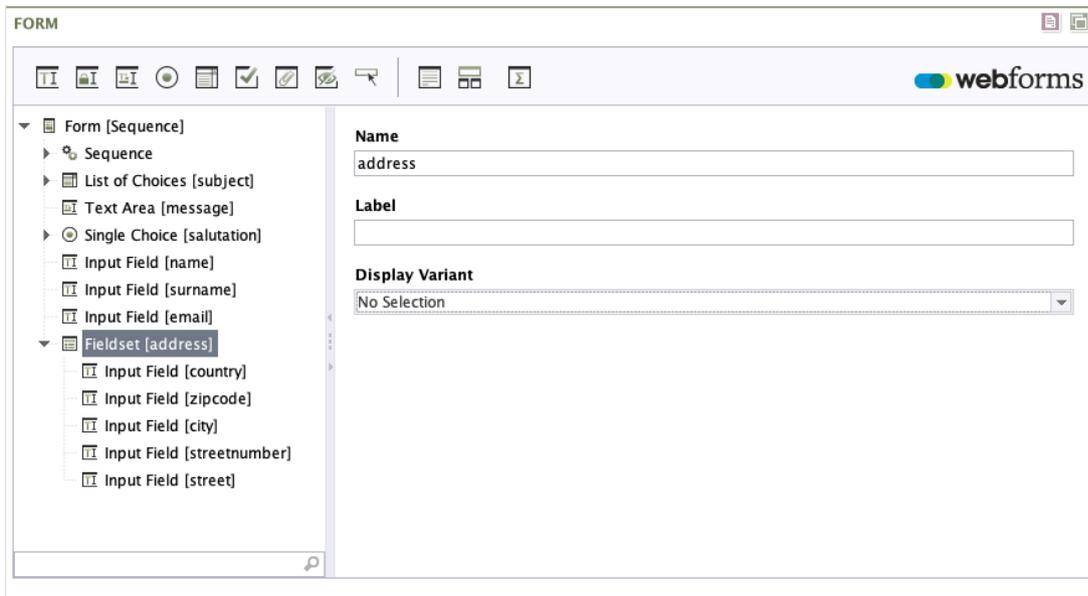


Figure 3.25. Layout

Name: Enter the technical name of the *Fieldsets* here. The name can be used by the browser to fill the input fields contained in the fieldset with data that the user previously entered into a form.

Label: You can use this field to enter an optional piece of label text.

Display Variant: Use this to pick an alternative presentation style for the Fieldset.

3.2.19. Layout

The Layout element consolidates form elements into layout groups. You can use this element to configure a two-column layout, for example. You can insert the Layout element at any position below the Form element. Just as with the Page Break element, the system assigns all of the form elements below the Layout to this layout.

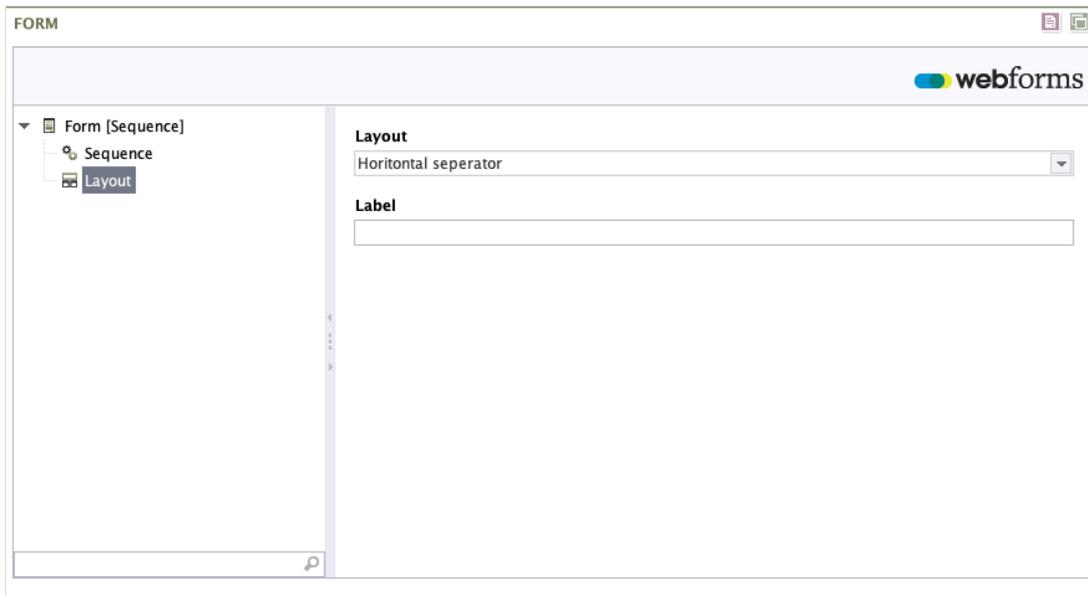


Figure 3.26. Layout

Layout: This field gives you the option of selecting a Layout variant. The Layout variants that are available are specified in the form's paragraph style sheet.

Label: You can use this field to enter an optional piece of label text. Whether (and where) the label is displayed in the form depends on the Layout selected.

3.2.20. Summary

This element presents a summary of all of the data entered into the form. You can, for example, create a "Thank you" page for the user as the last page of your form, while also summarising all of the form data the user has entered on this page. The user can then check the accuracy of the data here and correct his or her input as required. The user can use the "Back" button to go to the relevant form page and make any necessary changes.

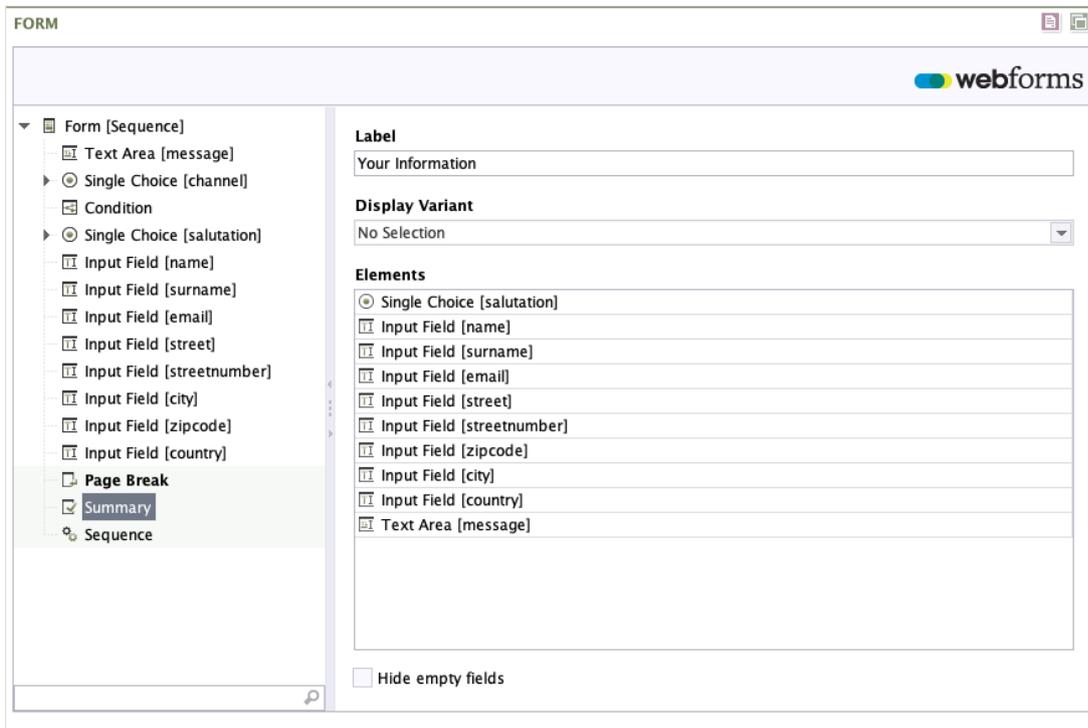


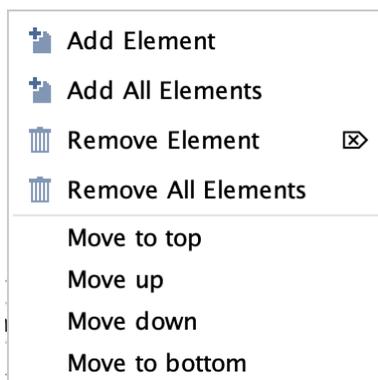
Figure 3.27. Summary

Label: Enter a piece of label text into this field; this is then displayed above the Summary.

Display Variant: Use this to pick an alternative presentation style for the Summary.

Elements: In this list, you specify the form values that will be displayed in the Summary. The values are output in the order in which they are arranged in the list.

To include a new form element in the Summary, first right-click the field with the mouse. In the menu that is then displayed, select the menu option *Add element*.



If you double-click the newly-added row, a drop-down menu is shown. From this list, select your chosen form element.

To include all existing form elements in the Summary, first right-click the field with the mouse. In the menu that is then displayed, select the menu option *Add all elements*.

If you do not make a specific selection here, then the default behaviour is for the user to be shown all values from the form elements selected for the Summary. If you want to include elements of type *Password* or *Hidden Field*, you have to make a specific selection here.

Hide empty fields: If you activate this check box, then only fields that have been filled out will be shown in the Summary.



Paragraph elements can be included in the Summary if you assign a name to the element beforehand.

3.3. Variables

You can draw on a range of variables when setting default values for input fields. These variables are replaced with a value when the form is displayed. As one example, you can set a field to have the current date as a default value by entering the variable ``${clientDate}` into the field's default value setting. Variables must always be specified using the format ``${name-of-the-variable}`.

All variables can be combined with additional text or other variables. The following variables are available to you as standard.

Variable	Description
date	The date in UTC on which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = "30/05/2013").
time	The time of day in UTC when the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 17:33).
serverDate	The date in the servers timezone on which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = "30/05/2013").
serverTime	The time of day in the servers timezone when the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 17:33).
clientDate	The date in the users timezone on which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = "30/05/2013").
clientTime	The time of day in the users timezone when the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 17:33).
timezone	The timezone configured for the user's browser (example: America/Los_Angeles).
language	The language configured for the user's browser. This is given in the form of the country code (de, en, etc.).
ip	The IP address assigned to the user's computer by his or her provider.

Variable	Description
remoteUser	Name with which the user logged in to your website. Please note: This variable is available only if the login was handled by the deployed Application Server.
principal	Name of the authenticated user (from the Principal object). Please note: This variable is available only if the login was handled by the deployed Application Server or JAAS is being used.
userAgent	Identification string supplied by the user's browser.
referer	The URL used to access the web page containing the form (also known as the referrer page or the history page in browser jargon). A relative or absolute URL is given as the value.

As an example, if you want to use the date and time to set a default value for a field, then you can achieve this by entering the following in the *Value* field:

Input: `${clientDate} ${clientTime}`

Output: 30/05/2013 18:36

3.4. Actions

You can select an action to define the type of procedure used to process the form data. Each form can contain only a single action. If you need to execute multiple actions per form, then this can be achieved by using a Sequence action (see Section 3.4.9, “Sequence”).

The form action is executed when the user clicks the submit button. For multi-page forms, this is located on the last page of the form. Accordingly, the position at which an action is inserted below the form element has no effect on the point in time at which it is executed.

For all actions (with the exception of the sequence action), you have the option of specifying the user input where the action is executed or not executed, as appropriate. As one example, this functionality could be used to send the user a copy of their form input via email if they have agreed to this as part of filling out the form.

To use this functionality, switch to the *Condition* tab in the detail view for the respective action.

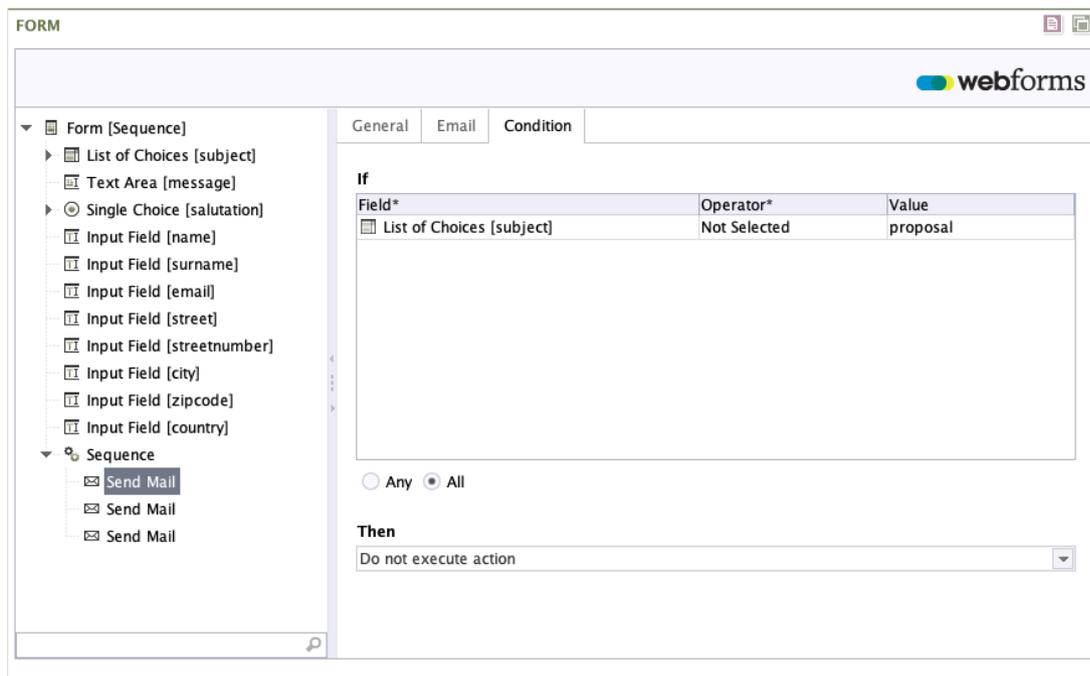


Figure 3.28. Condition tab

If: Use this field to specify the user input for which this action should not be executed/not executed.

The condition can be created as described in Section 3.2.11, “Condition”.

All of the other configuration options for form actions are described in detail in the following sections.

Then: In this drop-down list, you can specify whether the action is executed or not executed if the abovementioned condition is fulfilled.

3.4.1. Send as Email

The Send as Email action lets you send the form data entered by the user to any number of users (email addresses).

General tab

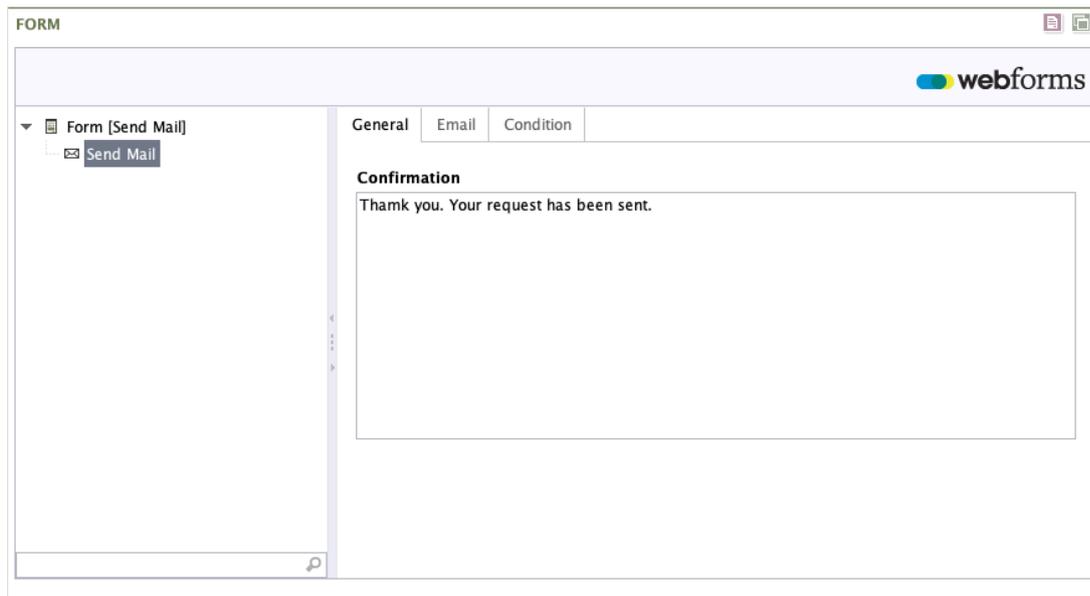


Figure 3.29. Send as Email – General tab

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Email tab

Enter the recipient, a subject line and the message body – just as you would when composing an email normally. In addition, you also have the option here of specifying which of the data items entered by the user should actually be sent in the email.

When configuring the Send as Email action, you can also utilise the data entered by the user directly – so as to send a copy of the message to the user, for example. To do so, simply enter a placeholder using the format `${element-name}` into the corresponding property field in the Send as Email action.

Example: In a contact form, you have defined an input field *email*, into which the user must enter his or her email address. If you now enter the variable `${email}` into the *CC* field, the user will receive a copy of the email created.

You can re-use the labelling for an option by extending the field name with the suffix *#label*, like this: `${element-name#label}`.

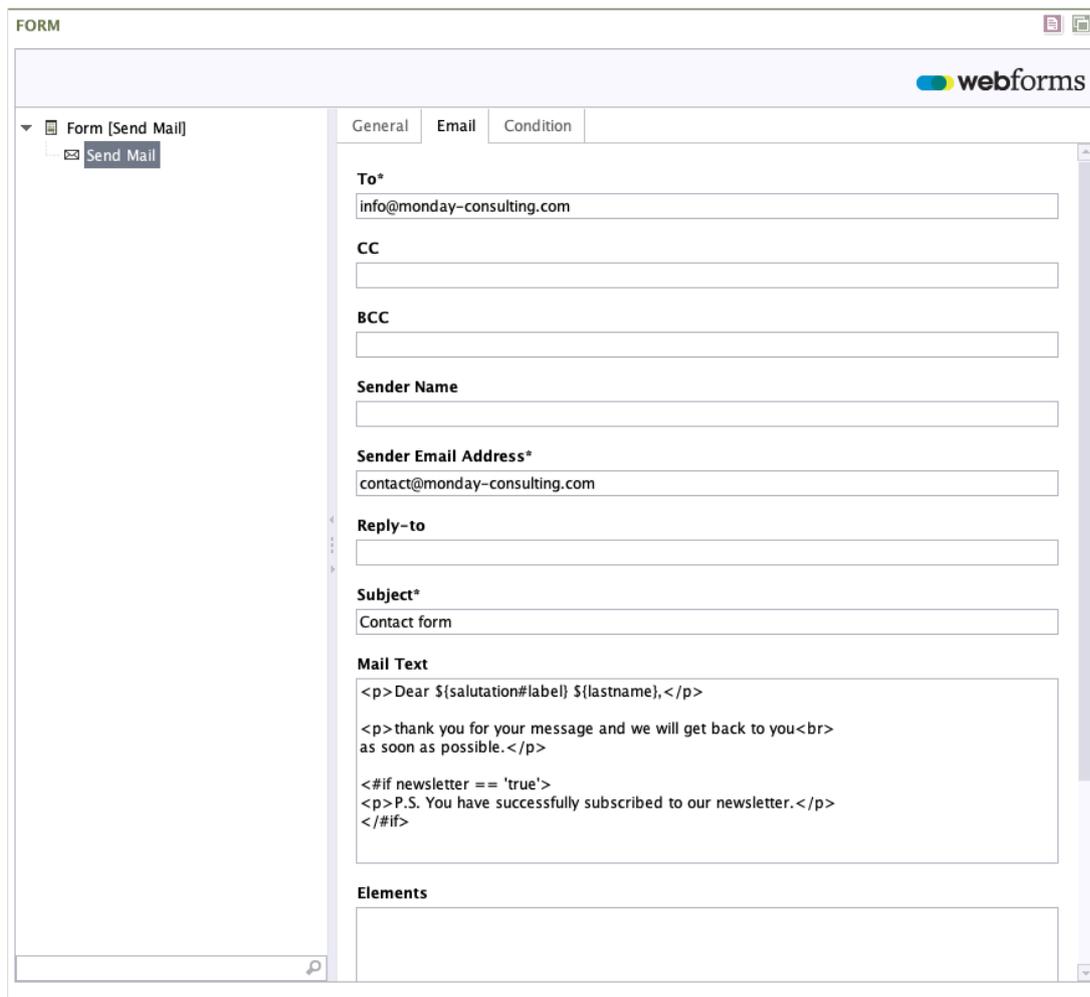


Figure 3.30. Send as Email – Email tab

You can send the email created to multiple recipients at the same time by entering multiple addresses (To, CC, BCC) into one address field. Use a comma to separate the individual addresses.

To: The email addresses to which the data entered by the user should be sent.

CC: The email addresses to which a copy of the data entered by the user should be sent.

BCC: The email addresses to which a blind carbon copy of the data entered by the user should be sent.

Sender Name: The name to be used as the sender.

Sender Email Address: The email address to be used as the sender.

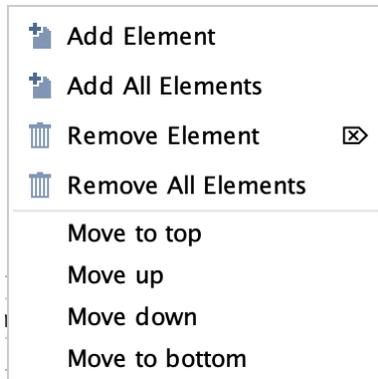
Reply-to: One or more comma-separated email addresses, to which any replies to the email should be sent (if not the same as the “Form” field).

Subject: The subject line of the email to be sent.

Mail Text: The text that is inserted into the body of the email, in addition to the form data.

Elements: In this list, you specify the form values that will be displayed in the email. The values are output in the order in which they are arranged in the list.

To include a new form element in the email, first right-click the field with the mouse. In the menu that is then displayed, select the menu option *Add element*.



If you double-click the newly-added row, a drop-down menu is shown. From this list, select your chosen form element.

To include all existing form elements in the email, first right-click the field with the mouse. In the menu that is then displayed, select the menu option *Add all elements*.

If you do not make a specific selection here, then the default behaviour is for the user to be shown all values from the form elements selected for the email. If you want to display fields of the *Password* or *Hidden Field* type, then you must make a specific selection.

Fields that have been hidden by conditions are never included in the email.

 *Paragraph* elements can be included in the email if you assign a name to the element beforehand.

Hide empty fields: By checking this box, you can specify that empty fields are not sent along with the generated mail.

Format: Select the predefined format option that should be used to format the created email. Webforms offers you a choice of the following format options as standard:

Format	Description
Text	Creates a plain text mail with the specified message text and the selected form values. The form values are added automatically as a simple list (label : value) at the end of the message text.

Format	Description
	<div data-bbox="400 248 1238 1043" style="border: 1px solid #ccc; padding: 10px;"> <p style="margin: 0;">From Sample Company <noreply@sample.com> Subject Contact Form To contact@sample.com</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Message</p> </div> <p>The following message has been sent via the contact form:</p> <p>Your questions to us: I have to say I'm very pleased with the service I've received so far. Keep the good work. Regards from a happy customer.</p> <p>Salutation: Mr. First name: John Last name: Smith E-mail address: john.smith@sample.com</p> </div>
HTML	<p>Creates an HTML mail with the specified message text and the selected form values.</p> <p>The form values are added automatically as a simple list (label : value) at the end of the message text.</p> <div data-bbox="400 1274 1238 1962" style="border: 1px solid #ccc; padding: 10px;"> <p style="margin: 0;">From Sample Company <noreply@sample.com> Subject Contact Form To contact@sample.com</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Message</p> </div> <p>The following message has been sent via the contact form:</p> <p>Your questions to us: I have to say I'm very pleased with the service I've received so far. Keep the good work. Regards from a happy customer.</p> <p>Salutation: Mr. First name: John Last name: Smith E-mail address: john.smith@sample.com</p> </div>

Format	Description
FreeMarker (text)	<p>Creates a plain text email.</p> <p>When this format option is selected, the message text is interpreted and executed as a FreeMarker template. With this format option, the form values must be added manually to the message text.</p> <pre data-bbox="359 456 1283 801"> Dear \${salutation#label} \${lastname}, thank you for your message and we will get back to you as soon as possible. <#if newsletter == 'true'> P.S. You have successfully subscribed to our newsletter. </#if> Best Regards </pre> <div data-bbox="400 837 1240 1384" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">From Sample Company <noreply@sample.com> Subject Contact Form To john.smith@sample.com</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Message</p> </div> <p>Dear Mr. Smith,</p> <p>thank you for your message and we will get back to you as soon as possible.</p> <p>P.S. You have successfully subscribed to our newsletter.</p> <p>Best Regards</p> </div>
FreeMarker (HTML)	<p>Creates an HTML email.</p> <p>When this format option is selected, the message text is interpreted and executed as a FreeMarker template. With this format option, the HTML code and form values must be added manually to the message text.</p> <p>The following example shows how you can modify the message text with the help of <code>#if</code> statements.</p> <pre data-bbox="359 1720 1283 1995"> <p>Dear \${salutation#label} \${lastname},</p> <p>thank you for your message and we will get back to you
 as soon as possible.</p> <#if newsletter == 'true'> <p>P.S. You have successfully subscribed to our newsletter.</p> </#if> </pre>

Format	Description
	<div style="border: 1px solid #ccc; padding: 10px; margin: 10px;"> <p style="text-align: center;">From Sample Company <noreply@sample.com> Subject Contact Form To john.smith@sample.com</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Message</p> </div> <p>Dear Mr. Smith,</p> <p>thank you for your message and we will get back to you as soon as possible.</p> <p>P.S. You have successfully subscribed to our newsletter.</p> <p>Best Regards</p> </div>

Webforms offers you the option of sending multiple emails with different content. To do so, simply create another Send as Email action. If you want to send the exact same content again, simply enter multiple recipients for the mail.

3.4.2. Excel File

This action saves the form data in a Microsoft Excel file, which is stored on the server. A new row is added to the file every time the form is submitted.

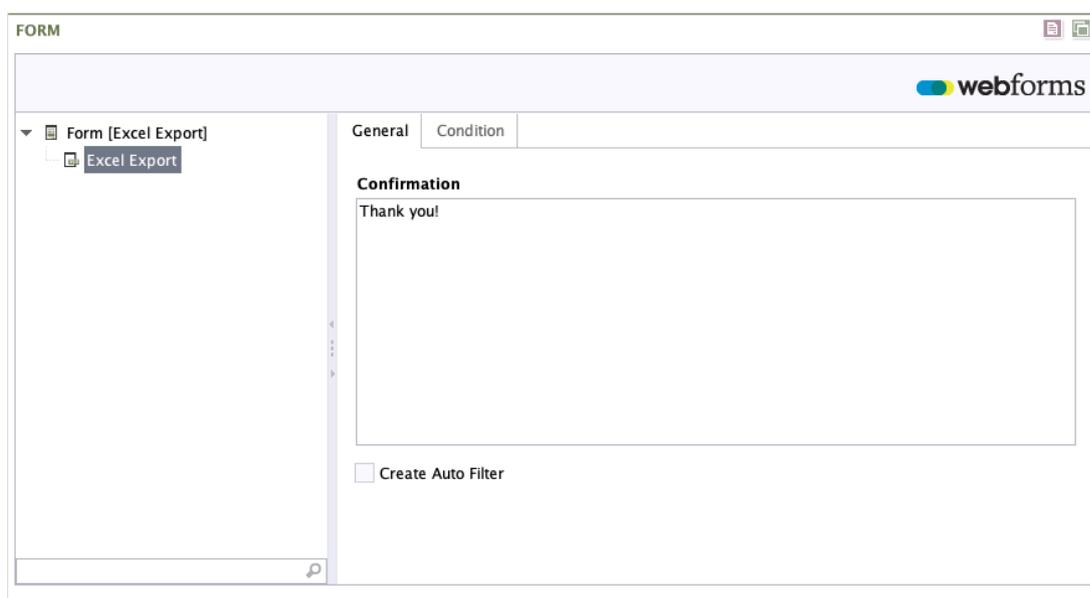


Figure 3.31. Excel File action – Properties tab

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Create Auto Filter: By activating this check box, you can specify that an AutoFilter should be created for every column within the Excel file created.

3.4.3. PDF

This action enables you to fill a PDF document containing interactive or editable form fields with the form data from your online form. The PDF document can then be downloaded and printed out by the end user.

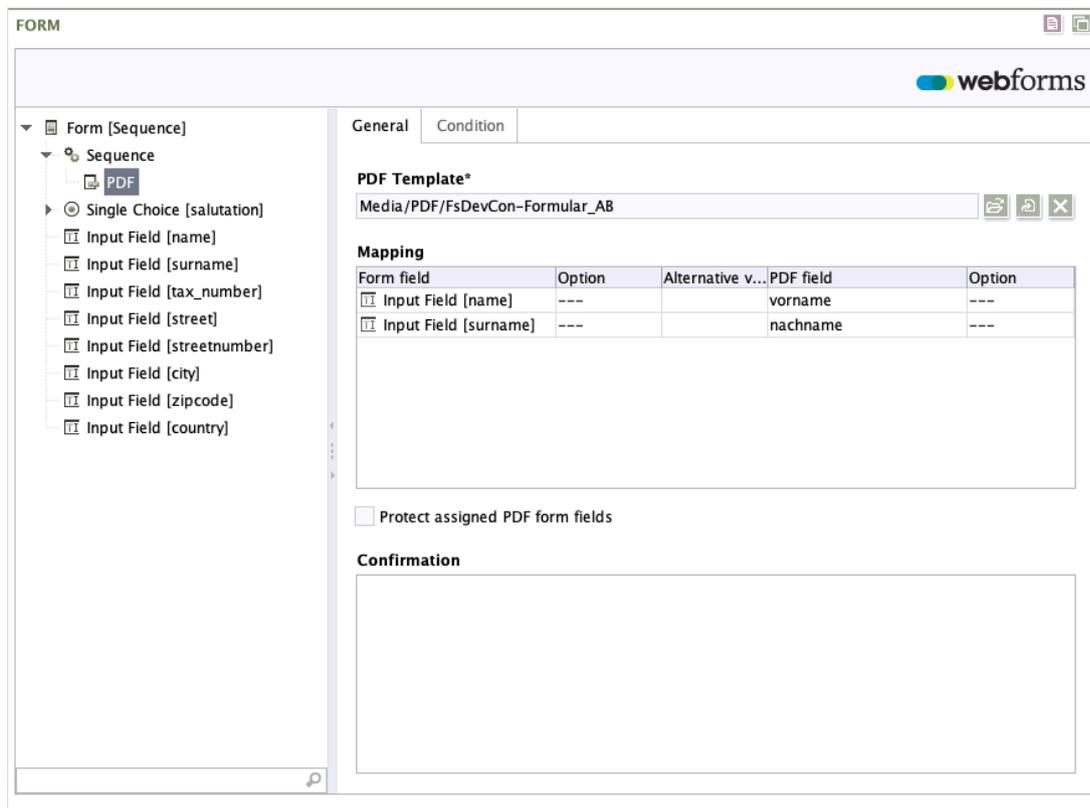
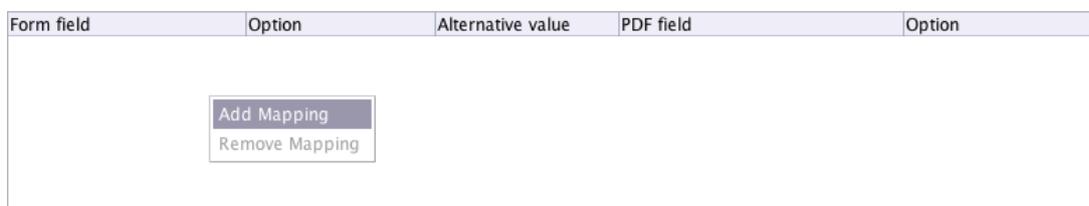


Figure 3.32. PDF action - Properties tab

Template: In this field, you can select a PDF document (PDF template) from the FirstSpirit Media Management.

Mapping: The list of mappings between the fields of the online form and the fields in the PDF form.

To add a new mapping to the list, right-click with the mouse on the assignment field.



If you double-click the row added, a drop-down menu is shown. From this list, select the form element that you wish to include in the assignment.

Form field	Option	Alternative value	PDF field	Option
<input type="text"/> Input Field [name] <input type="text"/> Input Field [surname] <input type="text"/> Input Field [tax_number] <input type="text"/> Input Field [street] <input type="text"/> Input Field [streetnumber] <input type="text"/> Input Field [city] <input type="text"/> Input Field [zipcode] <input type="list" value="List of Choices [marital_status]"/>				

If the form field selected is itself a selection field (radio button select field, check box select field or a drop-down list), then the column *Option* lets you specify a single option for this select field.

Form field	Option	Alternative value	PDF field	Option
<input type="list" value="List of Choices [marital_status]"/>	No Selection unwed married divorced widowed			

You can enter an alternative value in the *Alternative value* column. This will be entered into the PDF form instead of the value entered or selected by the end user.

Form field	Option	Alternative value	PDF field	Option
<input type="list" value="List of Choices [marital_status]"/>	married	<input type="text"/>		

You can select a form field from the PDF template in the *Field in the template* column.

Form field	Option	Alternative value	PDF field	Option
<input type="list" value="List of Choices [marital_status]"/>	married		<input type="text"/> Telefon Angabe freiwillig DienstAmtsbezeichnung Beschäftigungsdienststelle Anschrift Steueridentifikationsnummer ledig verheiratet seit verwitwet seit	

If the field selected is itself a selection field you can use the last column *Option* to select an option.

Form field	Option	Alternative value	PDF field	Option
<input type="list" value="List of Choices [marital_status]"/>	married		verheiratet seit	No Selection On

Protect assigned PDF form fields: If you activate this check box, then the user will be prevented from changing any assigned data within the PDF form.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Link text: The text for the download link, which the end user can click to download the generated PDF document. If you do not enter anything here, the name of the PDF file is used as the link text.



Before you can execute the PDF action, you must publish the selected PDF template, ensuring thereby that the template is available both in the Preview and the Live website.

To publish PDF templates, a special publication schedule must be both created and configured (see chapter 4.4 of the Developer Manual). Available publication schedules can be accessed via the menu option “*Projects > Execute schedule*”.

3.4.4. Data source

This action saves the form data in a FirstSpirit data source.

Form field	Datasource field
<input type="text"/> Input Field [name]	Firstname
<input type="text"/> Input Field [surname]	Lastname

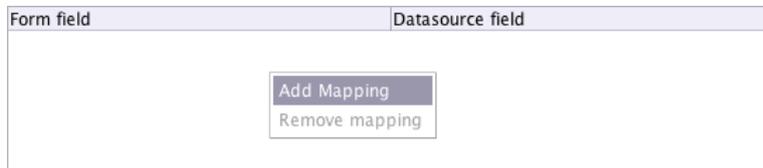
Figure 3.33. Datasource action – Properties tab

Schema: The schema in which the data source table is located.

Table: The data source table.

Mapping: The list of mappings between the form field and the data source.

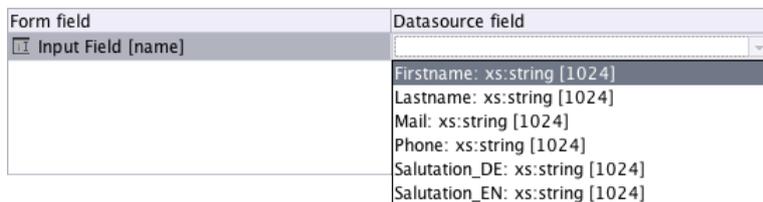
To add a new mapping to the list, right-click with the mouse on the mapping field.



If you double-click in the row that has just been added, a drop-down menu is shown, from which you can select the input element that you wish to include in the list.



In the *Data Source Field* you can select a field from the data source into which the value from the form should be saved. The select field displays only the kinds of data source fields that are compatible with the type of the form field selected in the first column.



Comment: The comment that should be added to the data record when it is saved. This comment is then visible in the record's version history.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, "Markdown").

Release stored data: If you check this box, the stored data record is automatically released in the FirstSpirit server.

3.4.5. Media Management

This action stores uploaded files in the FirstSpirit Media Management.

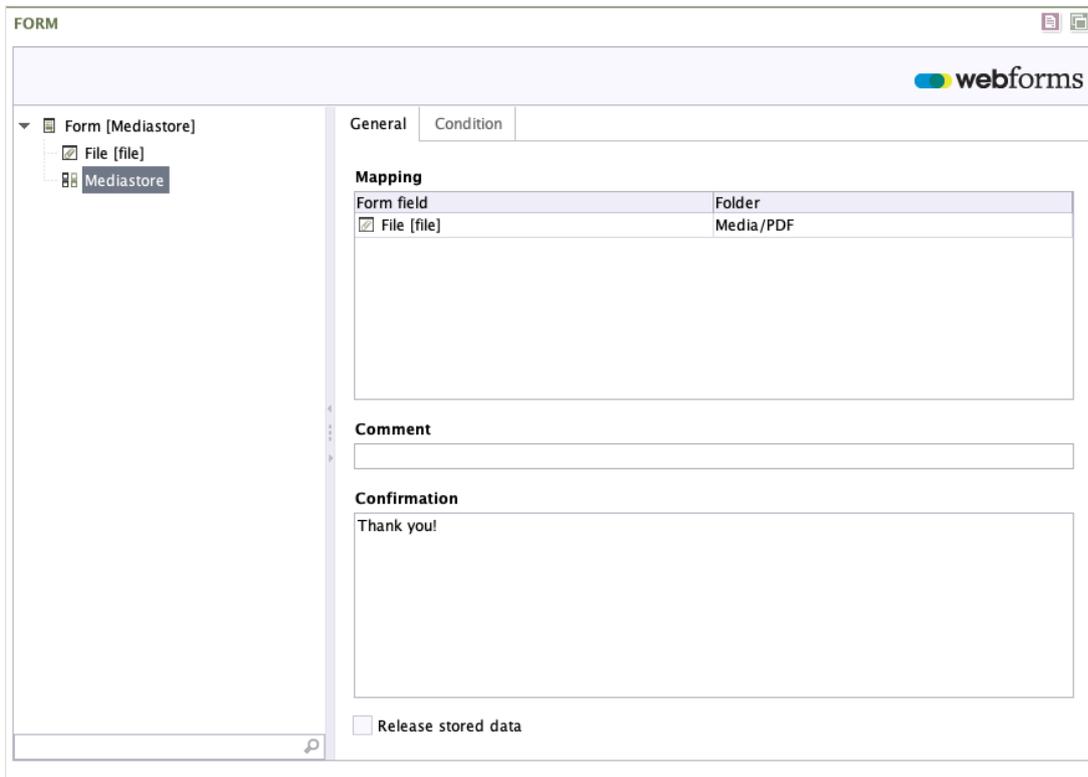


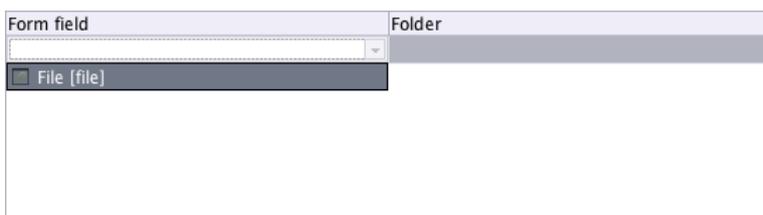
Figure 3.34. Media Store action – Properties tab

Mapping: The list of assignments between the file form field and a directory within Media Management.

To add a new assignment to the list, right-click with the mouse on the assignment field.



If you double-click in the row that has just been added, a drop-down menu is shown, from which you can select the file input element that you wish to include in the list.



In the *Directory in Media Management* column, you can select a directory from Media Management into which the uploaded file from the form should be saved. Double-click the column to open the selection dialog box.

Form field	Folder
<input checked="" type="checkbox"/> File [file]	Please select reference...

Comment: The comment that should be added to the file when it is saved. This comment is then visible in the version history and the media record description.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

Release stored data: If you check this box, the stored media record is automatically released in the FirstSpirit server.

3.4.6. Webforms Analytics

Use this action if you want to store and analyse the form data with the help of Webforms Analytics. For a detailed description, please consult the Webforms Analytics User Manual.

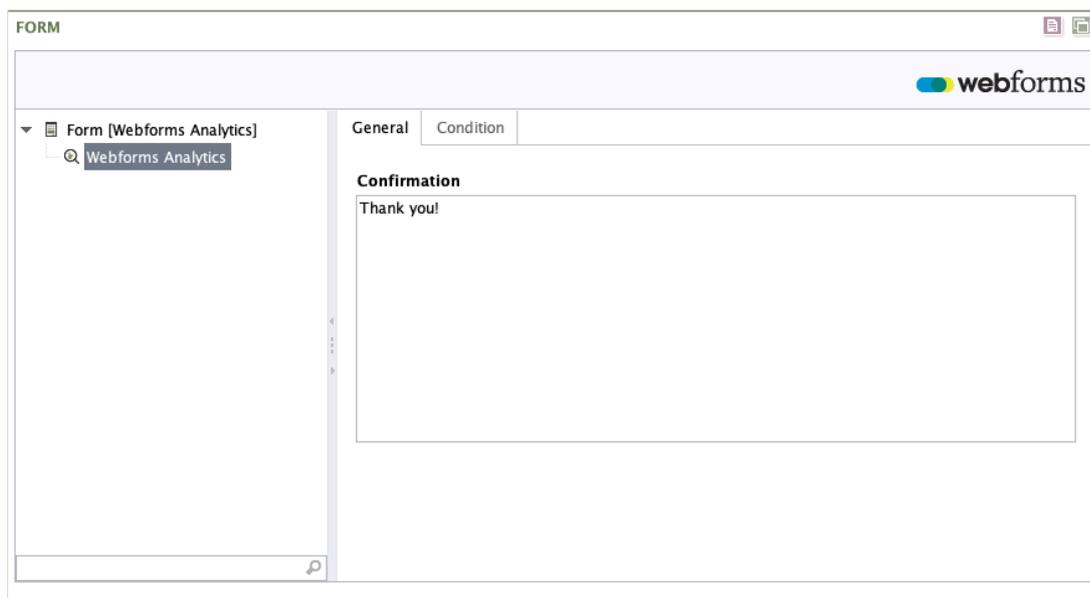


Figure 3.35. Analytics action – Properties tab

Confirmation: The confirmation message that is shown to the user on the webpage after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

3.4.7. Redirect

Use this action if you want to forward the user to a different page after submitting the form. You can reference an external address or a FirstSpirit page.

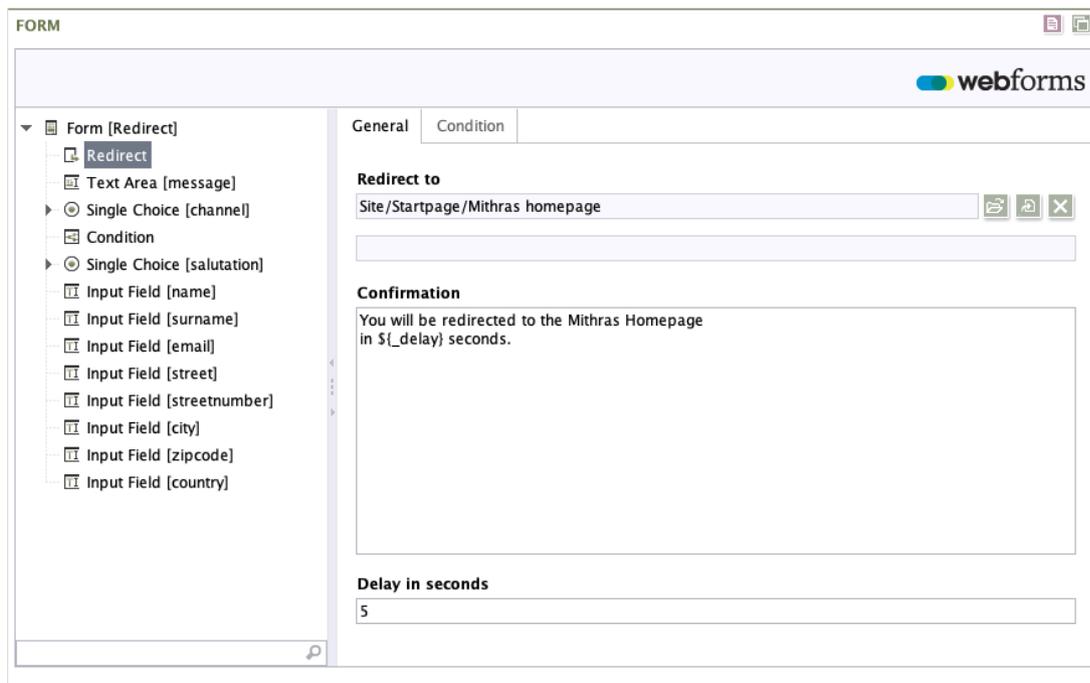


Figure 3.36. Redirect action – Properties tab

Forward to: In this field, specify the page to which the user will be forwarded after submitting the form. You can either select a page from the FirstSpirit content management system or you can specify an external URL.

As illustrated by the example shown below, you can also use Webforms variables in the external URL (see Section 3.3, “Variables”) as well as form values.

```
https://www.mydomain.com/site/${country}
```

Delay in seconds: Time in seconds before redirection to the target address.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. Apart from the form data, the variables `_url` and `_delay` are also available, which can be used to display the target address and/or the wait time. The confirmation text can also be formatted with the help of markdown (see Section 3.5, “Markdown”).

3.4.8. Webhook

You use the Webhook action to send form input directly to a specified URL or compatible web application as soon as a form has been completed.

This offers a straightforward approach to integrating third-party services such as Slack, Zapier or your own backend system.

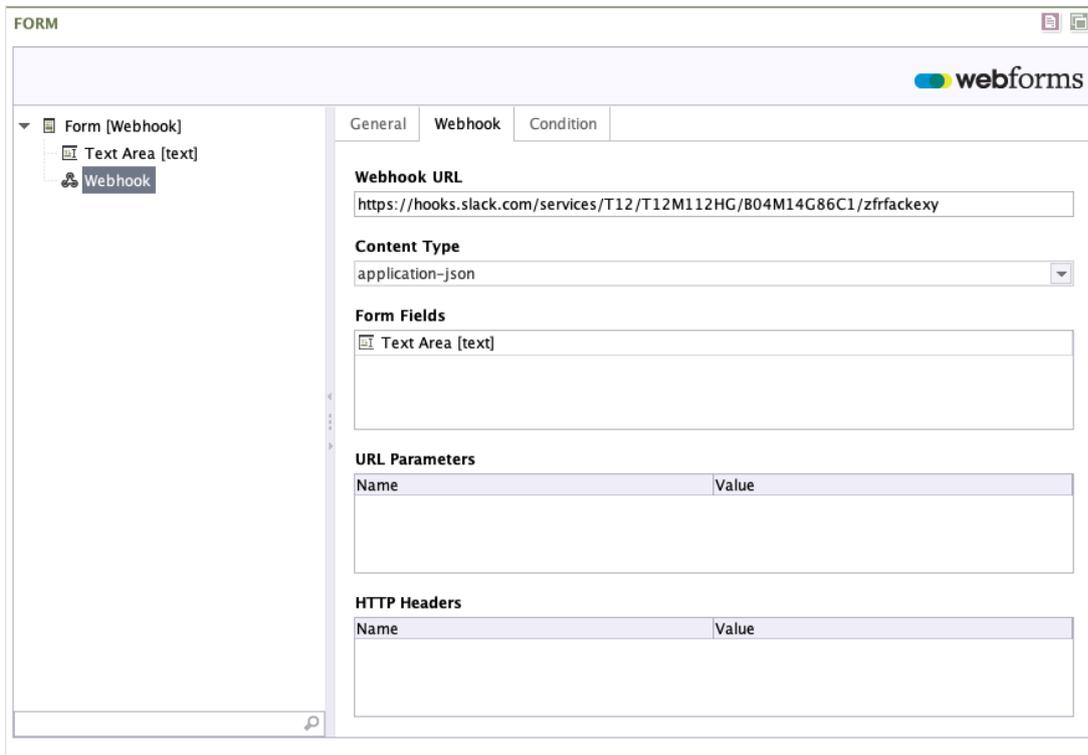


Figure 3.37. Webhook – Webhook tab

Webhook URL: Enter the URL to which the request should be sent.

The Webhook URL utilises the following format:

```
https://www.mydomain.com/path
```

The use of secure HTTP (HTTPS) is not mandatory but is strongly recommended. For security reasons, a local address (such as *localhost*, *127.0.0.1*, etc.) cannot be specified.

Content type: Select the format for the Webhook request. The following formats are supported:

Format	Description
application-json	Send the form data in JSON format in the body of the HTTP request.
application-x-www-form-urlencoded	Sends the form data as a URL-encoded data record separated by & characters in the body of the HTTP request.
multipart-form-data	Sends the form data as a multipart HTTP request. Use this content type if the form data to be sent contains file attachments.

Elements: Select the form fields whose data is to be sent to the Webhook endpoint.

URL parameters: Any additional parameters you want to append to the Webhook URL. When specifying parameter values, you can make use of form values and form variables by specifying the value as a placeholder with the format `${fieldName}` or `${variableName}`.

HTTP headers: You can specify user-defined HTTP headers that are to be used when sending the data to the specified Webhook endpoint. When specifying a header value, you can again use form values and form variables in the same way as when specifying the URL parameters.

3.4.9. Sequence

You can use the Sequence action to consolidate multiple actions into a single sequence. The individual actions are then executed in the order in which they are assigned below the Sequence action in the tree view. If an error occurs while the sequence is being processed, then the sequence stops at the action in which the error occurred.

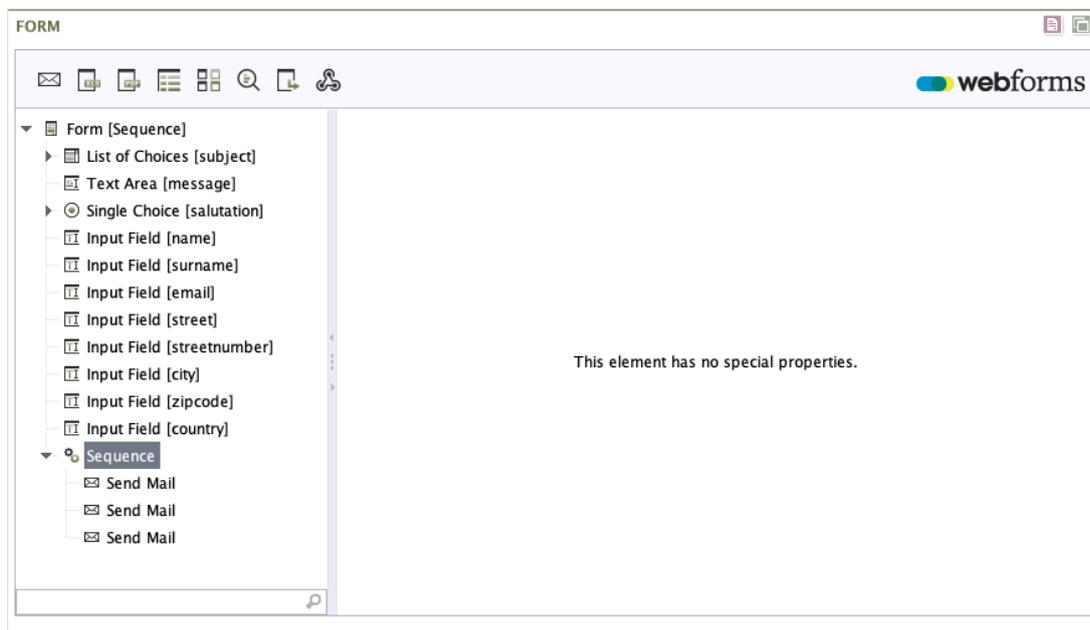


Figure 3.38. Sequence action

The Sequence action does not have any specific properties.

3.5. Markdown

Markdown is a simple mark-up language that you can use to add formatting and links to pieces of plain text. Markdown is available as standard for the following elements:

- Paragraph
- Note text
- Confirmation text

Examples of common kinds of markdown formatting are shown in the following table:

Formatting	Text as input	Text as displayed
Bold	Example for text in **bold type**	Example for text in bold type

Formatting	Text as input	Text as displayed
Italics	Example for text in <code>_italic type_</code>	Example for text in <i>italic type</i>
Numbered lists	1. Element 1 2. Element 2	1. Element 1 2. Element 2
Bulleted lists	* Element 1 * Element 2	• Element 1 • Element 2
Headings	# Heading level 1 ## Heading level 2 ### Heading level 3	Heading level 1 Heading level 2 Heading level 3
Links	[Linktext](http://sample-url.com) {param1=value1,...,paramN=valueN}	<u>Link text</u>

For full details of the formatting options available with markdown, please visit <https://commonmark.org/>.

A. Reserved identifiers.

For technical reasons, the technical name given to a form element must not match any of the following reserved identifiers:

`abstract, action, arguments, array, await, boolean, break, byte, case, catch, char, constructor, currentpage, currentpagenode, date, else, enum, eval, export, extends, false, final, finally, float, for, form, formdata, formvariables, function, goto, hasownproperty, if, implements, import, in, infinity, instanceof, int, interface, isfinite, isnan, isprototypeof, let, long, math, nan, native, new, nil, null, number, object, package, pagecount, pageelements, private, propertyisenumerable, protected, prototype, resolvedcaptchas, return, select-edelements, self, short, static, string, super, switch, synchronized, target, this, throw, throws, tolocalestring, tosource, toString, transient, true, try, typeof, undef, undefined, unwatch, valueof, var, view, void, volatile, watch, webforms_redirect, while, with, yield`