



# Creating Adobe® EchoSign® Forms with Text Tags

---

Version 6.0

Last Updated: May, 2014

© 2014 Adobe Systems Incorporated. All rights reserved.



## Table of Contents

- Overview ..... 4**
- Text Tag Version 2.0..... 4**
- EchoSign Text Tags Quick Start Guide..... 5**
  - Text Tag Examples .....5**
  - Conversion of Text Tags to EchoSign Form Fields .....6**
    - Field Length and Formatting..... 6
    - Interacting with Form Fields during Signing..... 7
    - Form Fields in the Signed document..... 7
  - Shortening Text Tags.....7**
- Adobe EchoSign Text Tags Deep Dive..... 8**
  - EchoSign Text Tags Syntax .....8**
  - Controlling Field Interactions during Signing .....9**
    - Assigning Form Field to Specific Signers ..... 9
    - Sender Prefills Data ..... 9
    - Unassigned Fields..... 9
  - Basic EchoSign Field Types.....10**
    - Signature and Initials Fields ..... 10
    - Special EchoSign Fields ..... 10
    - Special Read-only Fields ..... 10
    - Signature Block..... 11
    - Text Fields..... 11
  - Defining Required / Optional Fields .....11**
  - Editable / Read-Only Fields.....12**
- Advanced Document Processing With EchoSign Text Tags..... 12**
  - Additional Field Types.....12**
    - Check Boxes ..... 12
    - Radio Buttons ..... 12
    - Multi-Line Text Fields..... 14
    - Drop-Down Lists..... 14
    - Supporting Documents (File Attachments)..... 15
    - Hyperlinks..... 16
    - Participant Stamps and Transaction IDs ..... 16
  - Specifying Default Value .....17**
  - Specifying Tooltip .....17**
  - Specifying Font Appearance.....17**
  - Specifying Field Alignment.....18**
  - Specifying Repeat Fields.....18**
  - Specifying Data Validation Rules.....18**



- Supported Field validation rules ..... 19
- Masking Sensitive Data .....21**
- Specifying Conditional Rules For Fields.....21**
- Specifying Calculation Formulas For Fields .....23**
- Shortening Text Tags.....24**
- Removing Pages From Output ..... 25
- Advanced Text Tags Examples .....25**
- Check Boxes ..... 25
- Radio Buttons ..... 25
- Drop Down List..... 25
- Field Validation ..... 26
- Shortening Text Tags..... 26
- Conditional Rules ..... 26
- Calculated Fields ..... 27

## Overview

EchoSign Text Tags are specially formatted text that can be placed anywhere within the content of your document specifying the location, size, type of fields such as signature and initial fields, check boxes, radio buttons and form fields, and advanced optional field processing rules. Text Tags can also be used as names of PDF Form fields.

Text Tags are converted into EchoSign form fields when the document is sent for signature or uploaded to the EchoSign Document Library. EchoSign Text Tags can be placed in any document type such as PDF, Microsoft Word, PowerPoint, Excel and text files (.txt, .rtf).

For example:

- `{{Sig_es_:signer1:signature}}` – A signature field assigned to the first signer.
- `{{Cmpy_es_:signer1:company}}` – A field for the second signer to enter her company name.
- `{{address_es_:signer1}}` – A field for the first signer to enter her address.
- `{{*city_es_:signer1}}` – A mandatory field for the first signer to enter city.
- `{{!price}}` – A read-only field. Signers cannot change value of this field.

EchoSign Text Tags offer a powerful mechanism for setting up document templates that allow positioning signature and initial fields, collecting data from multiple parties within an agreement and define validation rules for the collected data. Once a document is correctly set up with Text Tags it can be used as a template when sending documents for signatures ensuring that the data collected for agreements is consistent and valid throughout the organization.

## Text Tag Version 2.0

This document defines the new syntax for EchoSign Text Tags that should be used for all new documents and new revisions of existing documents. Existing documents created using the older version of the Text Tag syntax will continue to work and be processed correctly by EchoSign.

We encourage migrating existing documents from the older version of the syntax to the new syntax to take advantage of the new capabilities. A single document can contain Text Tags defined with the old syntax and the new syntax; however a single tag **must** be defined entirely using the new syntax or the old syntax.

## EchoSign Text Tags Quick Start Guide

### Text Tag Examples

All the examples below assume tags are used within document content. When using tags as PDF form field names, omit the opening '{{' and the closing '}}'.

#### Signature and Initials fields

- `{{Sig_es_:signer:signature}}` – A signature field assigned to the signer.
- `{{Int_es_:signer1:initials}}` – An initials field assigned to the first signer.
- `{{SigB_es_:signer:signatureblock}}` – A signature block assigned to the signer.
- `{{OSig_es_:signer:optsignature}}` – An optional signature field assigned to the signer
- `{{OInt_es_:signer1:optinitials}}` – An optional initials field assigned to the first signer.

#### Special EchoSign fields

- `{{*Ttl_es_:sender:title}}` – A mandatory field for the sender to enter her title.
- `{{Cmpy_es_:signer1:company}}` – A field for the first signer to enter her company name.
- `{{N_es_:signer2:fullname}}` – A field for the second signer to enter her full name.
- `{{Dte_es_:signer:date}}` – A read-only field containing the date of signature.
- `{{Em_es_:signer:email}}` – A read-only field containing the signer's email address.

**Note:** The "signer" role specifies the first participant (Signer or Approver) who is not the Sender. The "signer1" role specifies the first participant in all cases. "signer1" could be the Sender if the Sender is signing or approving the document first.

The "sender" role is always specifies a field assigned to the Sender regardless if the Sender signs first or last.

Instead of using "sender" and "signer" roles, another approach is to only use "signer1", "signer2", "signer3" roles. Just make sure the participants are added in the correct order when sending the document.

#### Prefill

- `{{price_es_:prefill}}` – A field the sender fills in before sending the document.

#### Text fields

- `{{address_es_:signer1}}` – A optional field for the first signer to enter her address.
- `{{*city_es_:signer1}}` – A mandatory field for the first signer to enter city.

#### Read-only fields

- `{{!price}}` – A read-only field. Signers cannot change the value of this field.

#### Check Boxes

- `{{[]}}` – A check box.
- `{{[x]}` – A check box, checked by default.

### Radio Buttons

- `{{(Red)Color}}` – A radio button that has the value of Red when selected.

### Field Validation

- `{{*Mobile_es_:sender:phone}}` – A mandatory phone field allowing US phone number format
- `{{d_es_:signer:num(>0,<60)}}` – An optional number field allowing values between 0 and 60

### Conditional Rules

- `{{field_es_:showif(price>60)}}` – A field that will show if the field "price" has a value greater than 60.

### Calculated Fields

- `{{TV_es_:calc([SubTotal]+[Tax])}}`  
– A field whose value gets automatically calculated as the sum of "SubTotal" and "Tax" fields.

## Conversion of Text Tags to EchoSign Form Fields

### Field Length and Formatting

When using Text Tags in an editable document, EchoSign overlays a rectangle covering the entire span of the Text Tag (from the beginning "{" to the ending "}") and converts the Text Tag into its corresponding EchoSign Form Field.

The length of the Form Field can be increased by adding white space characters at the beginning of the Text Tag immediately after the "{" or at the end of the Text Tag just before the "}". Decreasing the size of the Form Field can be achieved by reducing the font size of the Text Tag to get the appropriate length.

The form field formatting (font size, type, color, etc.) is determined by the format of the first '{'. To ensure correct processing of Text Tags by EchoSign tag definitions should be specified in commonly occurring fonts within the document (Helvetica, Times New Roman, Arial, Verdana or Courier). Text Tag definitions are case sensitive and must be specified in lowercase text.

Figure below shows Text Tags placed within a Microsoft Word Document.

This Agreement is made effective as of `{{signerDate_es_:signer:date}}`, between My company ('Company') and `{{contractor_es_:signer:company}}` ('Contractor') `{{init_es_:signer:initials}}`

The entire span of the Text Tag from the beginning curly brace to the closing curly brace **must fit** within a single line. If a Text Tag span multiple lines either in the source document or in the processed PDF, the form field will not be correctly recognized by EchoSign and may cause the underlying text to be visible to the signer when signing the document.

When used within PDF Forms, the curly braces should not be included in the name of the PDF form fields and the size of the field is determined by the size of the PDF form field and formatting of the data entered into the field is determined by the PDF form field properties.



This Agreement is made effective as of , between My company ('Company') and  ('Contractor')

A single document cannot include both PDF Form fields and fields defined using Text Tags.


### Interacting with Form Fields during Signing

These Text Tags would be converted to the following signing experience

This Agreement is made effective as of Jun 14, 2012, between My company ('Company') and  ('Contractor')

### Form Fields in the Signed document

After data is merged into the signed document, the following would appear in the signed PDF

This Agreement is made effective as of Jun 14, 2012, between My company ('Company') and Contractor Co ('Contractor') 

### Shortening Text Tags

When used in editable documents, Text Tags containing many directives may become very long, making it difficult to specify small fields on the document. EchoSign provides a simple mechanism allowing you to specify shorter tags in the document.

For example, if you need to have a mandatory phone field assigned to the sender, instead of putting the full `{{*Mobile_es_:sender:phone}}` tag, you may use a shorter tag `{{ $mob }}` in the appropriate location in the document with whitespace to adjust the size of the field. The complete definition of the Text Tag `{{#mob=*Mobile_es_:sender:phone}}` can be placed at another location in the document.

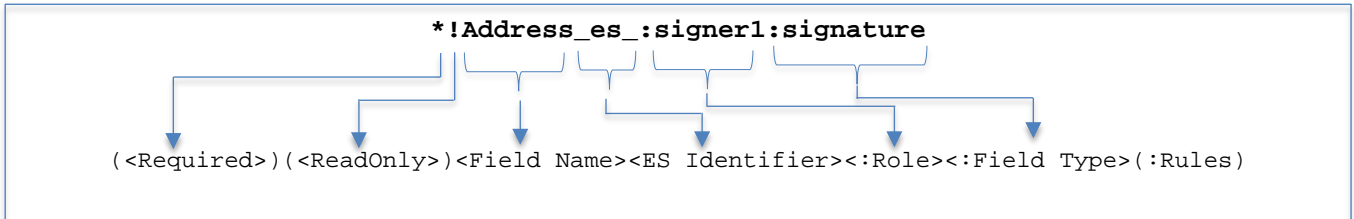
To learn more about Text Tag shortening, click [here](#).



## Adobe EchoSign Text Tags Deep Dive

### EchoSign Text Tags Syntax

An EchoSign Text Tag is defined as:



where:

- **Required** – is an optional flag that specifies whether the particular field is required and must be filled by the signer during signing. If the flag is not present the form field is not mandatory and the signer can (but is not required to) fill in a value during signing.
- **ReadOnly** – is an optional flag that indicates that the particular field is a read-only field. Marking a field as read-only ensures that the data in the field cannot be modified by anyone during the signing process. Data can be merged into read-only fields via the EchoSign API before the document is sent for signature.
- **Field Name** – is a unique name to identify this specific field within the document. The field name is used to identify the values that are provided during the signing process when the data gets exported from the service.

Note 1: If a document has more than one text tag with the same name but different directives, or those directives are in a different order amongst the same-name text tags, EchoSign will automatically add "efield" + unique number + "\_" to the beginning of the name. For example if two fields called "Address" appear in the document, EchoSign will change the name of one of the fields to "efield2\_Address".

Note 2: If there are two Text Tags with the same Field Name and same directives and order of directives within a document, these fields are treated as cloned fields. Data entered in one field is automatically cloned to the other field location within the document.

- **ES Identifier** – is a specific text string "\_es\_" which **must** be included after the Field Name in the Text Tag in order to enable EchoSign specific features. If the EchoSign identifier is not included after the field name a form field will be added to the document without any directives.
- **Directive** – A list of one or more definitions beginning with a colon (":") that specify the Role, the Field Type, and additional field processing Rules any conditional semantics associated with the field and any validation rules defining the type of input allowed in the field. Directives can be defined in any order.
  - **Role** – the participant this field is associated with (eg: sender, signer, etc)
  - **Field Type** – the type of EchoSign form field (eg: Signature, initials, check box or a text field)





- Rules – Advanced processing rules such as data validation, logic to show/hide fields based on conditions, etc.

## Controlling Field Interactions during Signing

Form fields in a document can be assigned to a specific signer during the signature workflow. EchoSign will ensure that a signer can only interact with the fields that have been assigned to him/her during the signature process. Form fields can be assigned to a specific signer by adding a specific directive in the Text Tag.

### Assigning Form Field to Specific Signers

EchoSign supports the following directives for assigning form field to specific signers.

<code>Field1_es_:sender</code>	A field in the document assigned to the sender of the document.
<code>Field2_es_:signer</code>	A field in the document that is assigned to the first signer that is not a sender of the document.
<code>Field3_es_:signer1</code> <code>Field4_es_:signer2</code>	If a document requires more than one signer and each signer needs to provide data during signing, fields in the document can be assigned to specific signers using <code>signer&lt;x&gt;</code> directive where <code>&lt;x&gt;</code> is order in which the signers sign the document. Fields marked with <code>signer1</code> directive are assigned to the first signer, fields marked with <code>signer2</code> directive are assigned to the second signer, and so on. EchoSign will ensure that a signer can only interact with the fields that are assigned to her. If the sender of the document signs first during the signature process, any fields marked as "signer1", in addition to the field marked as "sender", will be presented to the sender.

### Sender Prefills Data

In some cases, the sender is not required to sign the document but is required to enter data before the document is sent out for signature. This is accomplished by adding a "prefill" directive to the relevant form fields within the document. EchoSign will allow the sender to complete those fields before sending the document for signature.

<code>Field5_es_:prefill</code>	A field in the document that needs to be filled by the sender before the document is sent for signature.
---------------------------------	--

**Note:** Prefill is enabled by default only for documents uploaded to your EchoSign library or if the "Preview, position signatures or add form fields" option is selected when sending the document. Contact EchoSign support to enable always checking for prefill fields within a document sent directly from the EchoSign send page.

### Unassigned Fields

During the signature process any signer can fill a field that does not have a directive assigning it to a specific participant. After the field has been filled, it will be locked to prevent subsequent signers from changing its value.

<code>{{Address}}</code>	A field in a document that can be filled by any signer.
--------------------------	---

<code>{{zip_es_:zip4}}</code>	A field in a document for any signer to enter in a zip code with validation.
-------------------------------	--

## Basic EchoSign Field Types

A variety of field types can be created in a document using specific directives within Text Tags. EchoSign will interpret the tags and convert them to the appropriate form fields during the signature process. All the field types that can be added to a document through the EchoSign Web UI can also be specified through Text Tags.

### Signature and Initials Fields

You may position EchoSign signature or initials fields at specific locations within the document by adding a directive to designate the type of the signature field. By default, signature and initials fields are mandatory. However, some documents require that the signer optionally initial or sign at specific additional locations within the document. Optional signature or initials fields can also be defined within a document using Text Tags.

**Note:** EchoSign requires that a document contain at least one signature within the document for each signer of the document. Approvers of a document are not required to have a signature field on the document during the signature process.

<code>Sig1_es_:signature</code>	A required signature field assigned to the sender.
<code>Int1_es_:signer1:initials</code>	A required initials field assigned to the first signer.
<code>OSig1_es_:signer2:optsignature</code>	An optional signature field assigned to the second signer.
<code>OInt1_es_:signer1:optinitials</code>	An optional initials field assigned to the first signer.

### Special EchoSign Fields

EchoSign provides special fields that can be positioned within the document to collect information from the signer such as signer's job title or company name. If the information for these fields is already available EchoSign will automatically populate these fields when presenting to the signer during signing.

<code>Ttl1_es_:title</code>	A field to specify the signer's job title.
<code>Cmpy1_es_:company</code>	A field to specify the signer's company.

### Special Read-only Fields

In addition, EchoSign also supports collecting the date of signature, the email address and the name of the signer within the document. These fields are automatically populated by EchoSign and cannot be modified by the signer when signing the document.

<code>Dte1_es_:date</code>	A field to automatically capture the date of signature.
<code>Em1_es_:email</code>	A field to automatically capture the signer's email address. When used in EchoSign Widgets, the email address field is editable by the signer when he/she signs the Widget. Depending on your account configuration EchoSign will verify email address provided by the signer before completing the Widget signature process.
<code>Name1_es_:fullname</code>	A field to specify the signer's fullname.



## Signature Block

If no signature field is placed in the document for a specific signer, EchoSign will automatically place a signature block at the bottom of the document for the particular signer.

Depending on your account configuration, signers may also be required to provide their title and company. EchoSign will present a signature block containing those additional fields in case they were not already placed in the document.

You may position the entire EchoSign signature block in your document using the "signatureblock" directive.

<code>SBlk1_es_:signatureblock</code>	Positions a signature block within the document at the specified location.
---------------------------------------	--

## Text Fields

All fields defined within a document that do not contain any additional field type definition directives are treated as fillable text fields within a document.

<code>{{propertyName_es_:signer}}</code>	Defines an editable text field called "propertyName" within a document.
--	---

## Defining Required / Optional Fields

For most fields, you may specify if it is required or optional. EchoSign ensures that a signer cannot sign the document before all mandatory fields are filled.

By default, all fields are optional with the following exceptions:

- Signature and Initials fields are mandatory by default unless you define these fields using the `optsignature` or `optinitials` Text Tags.
- Title and Company fields are mandatory if your account preferences require them to be part of the signature block.

A mandatory field is marked with an asterisk (\*) prefix to its name or by adding the **required** directive in the Text Tag.

For example:

<code>{{*phone_es_:sender}}</code> <code>{{phone_es_:sender:required}}</code>	Defines a mandatory text field called "phone" to be filled by the sender.
<code>{{address_es_:signer}}</code>	Defines an optional field called "address" to be filled by the signer.

Required fields will be outlined in red when signing.

<b>Job Title:</b>	<input type="text" value="Please enter your job title..."/>
-------------------	---



## Editable / Read-Only Fields

By default, all fields are editable. Exceptions include some special signer fields (email, signer name, and signature date) that are read only by default. (See Special Read-only Fields).

Read-only fields are useful when automatically merging data into the document using the EchoSign API or with EchoSign's MegaSign Data Merge. Read-only fields ensure the signer cannot change the merged data when signing.

Adding an exclamation mark (!) prefix to the field name or using the **readonly** directive marks it as a read-only field.

For example:

<code>{{!price}}</code> <code>{{price_es.:readonly}}</code>	Defines a read-only text field to be filled in by merging data before the document is sent out for signature.
--	---

## Advanced Document Processing With EchoSign Text Tags

### Additional Field Types

#### Check Boxes

You can create a checkbox using Text Tags by placing the Check Box tag in your document or by using the checkbox directive.

<code>{{[]}}</code> <code>{{CB_es.:checkbox}}</code>	Positions an unnamed checkbox within a document.
<code>{{[x]}}</code> <code>{{CB1_es.:checkbox(checked)}}</code>	Positions an unnamed checkbox that is checked by default within a document.
<code>{{[]Comm_es.:signer1}}</code> <code>{{Comm_es.:checkbox:signer1}}</code>	Positions a checkbox called "Comm" assigned to the first signer.
<code>{{[*]}}</code> <code>{{CB2_es.:required:checkbox}}</code>	An unnamed mandatory checkbox that must be filled in by the first signer. For information about making fields mandatory see Section Defining Required / Optional Fields.
<code>{{[!]}</code> <code>{{CB3_es.:readonly:checkbox}}</code>	An unnamed read-only checkbox. For information about making fields read-only see Section Editable / Read-Only Fields.
<code>{{CB4_es.:checkbox:label("I agree.")}}</code>	Defines a checkbox with a label for a user-friendly description of the checkbox. The label is displayed to the signer during signing and in this case will display: "I agree."

#### Radio Buttons







Text tags can be used to define and position radio buttons within a document. Radio buttons must have at least two options (choices) within a radio button group – a group identified by the same field name within the document. Each option (choice) of the radio button must have a different selection value. Radio buttons can be defined by using the following syntax.

<code>{{(option_value)fieldname_es_}}</code>	Positions a radio button within a document.
--	---

<pre> {{{Red)Color_es_:signer1}} {{{Blue)Color_es_:signer1}} {{{Green)Color_es_:signer1}} </pre>	<p>Defines three radio buttons belonging to the same group called "Color" assigned to the first signer in the document. The options have values of Red, Blue, and Green.</p>
<pre> {{Color_es_:signer1:radio(Red)}} {{Color_es_:signer1:radio(Blue)}} {{Color_es_:signer1:radio(Green)}} </pre>	<p>An alternative way to define the Color radio button group using the radio directive.</p>
<pre> {{{Red)Color_es_:signer1:label("Red")}} {{{Blue)Color_es_:signer1:label("Blue")}} {{{Green)Color_es_:signer1:label("Green")}} </pre>	<p>Defines a radio button group and gives each radio button a user-friendly label. The label is displayed to the signer during signing.</p>

### Specifying Radio Button Style

EchoSign supports creating Radio Buttons in different styles. The specified Radio Button style is displayed in the PDF, during the signing the Radio Buttons options are always presented as circle.

Text Tag Directive	Permitted values		Example
:style	circle – Default style if none is specified.		:style(circle) :style(diamond) :style(star) :style(check) :style(cross) :style(square)
	diamond		
	star		
	check		
	cross		
	square		



## Multi-Line Text Fields

Text tags can be used to define text fields that can span multiple lines. These fields can be used within documents to collect multiple lines of data during the signing process.

<code>{{largeField_es_:signer1:multiline}}</code>	Defines a field within a document assigned to the first signer called largeField which will allow the signer to enter multiple lines of data in the field during the signature process.
<code>{{largeField2_es_:signer1:multiline(4)}}</code>	Defines a field within a document assigned to the first signer called largeField2 which will allow the signer to 4 lines of data during the signature process. If the number of lines parameter is missing the field will support 2 lines of data entered by the signer.

## Drop-Down Lists

Text tags can be used to define and position drop-down lists within a document. Drop-down lists can contain defining multiple options from which the user can choose one of the available option when signing the document. A drop-down list definition has two components: a unique set of options from which the user can select one choice and optionally a set of values that are returned when exporting the data from the agreement.

Drop-down options must be unique though drop-down values do not need to be unique i.e. multiple drop-down options can have the same value exported from them. Drop-down values are optional. If no drop-down values are defined, the drop-down options are returned when the data gets exported from the agreement. When drop-down values are defined the number of values defined **must** match the number of options. If the number of options and values specified do not match, this results in an error and the field will not be treated as a drop-down list.

When defining a default selection for a drop-down list or defining conditions for another field based on the user's selection of one of the option from the drop-down list, the export values (not the options) must be used. If there are no export values defined, then use the drop-down options.

<code>{{*Color_es_:signer1:dropdown(options="Red,Green,Blue")}}</code>	Positions a required drop-down list within a document. During signing the list will allow the user to select from one of the three available choices. This drop-down list does not specify any export values, as a result the option selected will be exported from the Manage Page, through the API or from the Reports.
--	---



<pre> {{Color_es.:signer1:dropdown(options="Red,Green,Blue", values="R,G,B")}} </pre>	<p>Positions a drop-down list within a document. During signing the list will allow the user to select from one of the three available choices. In this case the drop-down list specifies the values to be exported, as a result depending on the option selected by the user during signing the corresponding value gets exported from the Manage Page, through the API or from the Reports. For example if the user selected "Red" the value exported will be "R"</p>
<pre> {{Color_es.:signer1:dropdown(options="Red,Green,Blue", values="R,,B")}} </pre>	<p>In case the drop-down list options are the same as the previous example but one of the value is empty. In this example if the user selects "Green" as the option in the drop-down no value gets exported for this field.</p>
<pre> {{Color_es.:signer1:dropdown(options="Red,,Blue", values="R,G,B")}} </pre>	<p>This drop-down presents three options to the user when signing, with the second option being blank. If the user selects the blank option, the value exported will be "G".</p>

Additional directives can be added to the drop-down fields definition as needed. For example: making a drop-down list required, applying conditional logic to drop-down lists or specifying that the option selected by the user should be masked in the PDF.

### Supporting Documents (File Attachments)

EchoSign can be used to collect supporting documents from signers during the signing process. Supporting documents can also defined be as any other form field within a document using a Text Tag. During the signing experience, EchoSign will convert the Text Tag into a field where the signer will be able to select a file for upload. The uploaded document(s) are included as part of the signed document and are attached at the end of the signed agreement after the signature process. Form fields for collecting files from the signer are defined by using the "**attachment**" directive.

<pre> {{*DriversLicense_es.:signer1:attachment}} </pre>	<p>Defines a mandatory field called "DriversLicense" assigned to the first signer where the signer will be required to upload a file.</p>
<pre> {{photo_es.:signer1:attachment:label("Photo")}} </pre>	<p>Defines an optional field called "photo" assigned to the first signer where the signer can upload an image. The directive label can be used to specify a user-friendly</p>

	description for the field. The label is displayed to the signer during signing.
--	---

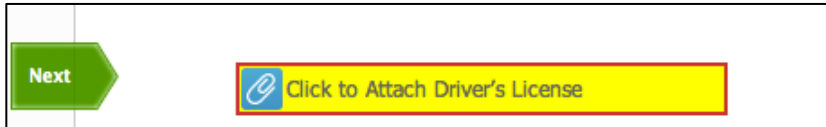
Defining a file attachment field:

```

{{*DriversLicense_es_:signer1:attachment:label("Driver's License")}}

```

File Attachment field presented during signing:



### Hyperlinks

EchoSign can be used to specify hyperlinks in documents. Hyperlinks are clickable during signing and can navigate the user to a location within the document or to an external URL. Labels can also be specified for hyperlinks so that the document displays the text label instead of the direct URL.

<code>{{companywebsite_es_:link(http://www.adobe.com):label(Adobe)}}</code>	Defines a hyperlink that links to an external URL. Includes a label "Adobe" that is displayed to the signer.
<code>{{doclink_es_:link:page(10):label(View this section)}}</code>	Defines a hyperlink that links to page 10 in the current document. Includes a label "View this section" that is displayed to the signer.

### Participant Stamps and Transaction IDs

Participant Stamps and Transaction IDs can be added to the signed PDFs for signers. The specific participant's stamp will appear after that person has signed or approved the document. The Participant Stamp will include the participant's name, signing/approval event time, and email. The Transaction IDs and stamps will appear in the final signed PDF.

<code>{{userstamp1_es_:signer1:stamp}}</code>	Defines a participant stamp for the first participant that will appear in the final signed PDF.
<code>{{userstamp2_es_:signer2:stamp}}</code>	Defines a participant stamp for the second participant that will appear in the final signed PDF.
<code>{{transstamp2_es_:transactionid}}</code>	Defines a transaction ID stamp that will appear in the final signed PDF.
<code>{{userstamp_es_:signer1:stamp:repeat}}</code> <code>{{transstamp1_es_:transactionid:repeat}}</code>	Defines a stamp that will appear in the same location on all pages in the document.





## Specifying Default Value

EchoSign supports specifying default value for a field using Text Tags. The specified default value is shown to the participants during the signing process.

Text Tag Directive	Example
:default	:default("Default value specified in double quotes") :default('Default value specified in single quotes')

If the default value contains a single quote or apostrophe, the default value must be enclosed within double quotes. For example: `{{location_es_:signer1:default("Signer's Business")}}`. If double quotes are required within the default value the default value must be placed within single quotes.

## Specifying Tooltip

Text Tags also allow specifying Tooltips for EchoSign form fields. Tooltips are displayed to the user during the signing process.

Text Tag Directive	Example
:tooltip	:tooltip("Please provide a copy of your Driver's License") :tooltip('Choose a color')

## Specifying Font Appearance

Text Tags can be used to change the font appearance of the field so that it overrides the font appearance used in the text tag for font name and size. To specify font name, use the font name of any font in the existing document or either of the following fonts: Helvetica, Times, Courier. By default the font name and size will be the font name and size of the first bracket "{" at the beginning of the text tag.

Text Tag Directive	Example
:font	:font(name=Times, size=12) – Specifies the field appearance to be Times font text and size 12 point font  :font(size=14) – Specifies the field appearance to be size 14 point font



## Specifying Field Alignment

Text Tags can be used to align the values entered in the form fields. By default the values in the form fields are left aligned. However using directives on the Text Tags, the form fields can be aligned to be center aligned or right aligned.

Text Tag Directive	Example
:align( <i>alignment</i> )	:align(left) – Aligns the text in the field to the left. This is the default behavior for form fields and the directive does not need to be satisfied. :align(right) – Aligns the data in the field to the right. :align(center) – Center aligns the data entered in the form field.

## Specifying Repeat Fields

Text Tags can be used to repeat the same fields on each page in the document without having to copy the text tag on each page. For example, this could be particularly useful for Signature Fields or Participant Stamps on lengthy documents.

Text Tag Directive	Example
:repeat	:stamp:repeat – Copies the participant stamp to the same location on each of the other pages in the document. Make sure to include a signer role for the text tag. (e.g. signer1, signer2) :signature:repeat – Copies the signature field to the same location on each of the other pages in the document. Make sure to include a signer role for the text tag. (e.g. signer1, signer2) :repeat(even) – Copies the field to the same location on the even pages in the document :repeat(odd) – Copies the field to the same location on the odd pages in the document :repeat(after) – Copies the field to the same location on all the pages after the page where the field is added :repeat(before) – Copies the field to the same location on all the pages before the page where the field is added :repeat(2,10), :repeat(6-9), :repeat(2,6-9) – Copies the field to the same location on the specified page or page ranges in the document

## Specifying Data Validation Rules

EchoSign supports a variety of field validation types specified by additional Text Tag directives. Specified data validation rules are applied when the signer is filling in the data during signing. EchoSign will not allow the user to complete signing the document if the data entered does not match the specified validation rule.



For example:

<code>{{*Mobile_es_:sender:phone}}</code>	Defines a required field called "Mobile" assigned to the sender where the sender must provide a valid phone number.
<code>{{Birthday_es_:signer2:isdate}}</code>	Defines a field called "Birthday" where the second signer needs to enter a date value.

Some field validation types may have additional parameters that narrow down the allowed values in the field.

<code>{{duration_es_:signer:num(&gt;=0,&lt;=60)}}</code>	Defines an optional field called "duration" where the value of the field must be between 0 and 60.
<code>{{tenNum_es_:signer1:string(char=num,maxlen=10)}}</code>	Defines an optional field called "tenNum" which allows entering up to a maximum of 10 digits.

### Supported Field validation rules

Validation Type	Text Tag Directive	Optional Parameters & Values	Samples	
String	:string		:string	
		char	alpha – Letters only	:string(char=alpha)
			num – Numbers only	:string(char=num)
			alphanum – Letters & numbers	:string(char=alphanum)
		maxlen	Specific maximum length	:string(maxlen=10) :string(char=alpha,maxlen=10)
Number	:num		:num	
		>=	Greater than or equal	:num(<=1000)
		<=	Less than or equal	:num(>=50,<=500)
Currency	:curr		:curr	
		country	us – Default	:curr(country=us)
			uk - United Kingdom	:curr(country=uk)
		>=	Greater than or equal	:curr(<=500)
	<=	Less than or equal	:curr(country=us,<=100)	
Percentage	:pct		:pct	
		>=	Greater than or equal	:pct(>=50)
		<=	Less than or equal	:pct(>=0,<=100)
Date	:isdate		:isdate	
		format	mm/dd/yy (Default)	:isdate(format=dd/mm/yy)
			mm/dd/yyyy	
			dd/mm/yy	
			dd/mm/yyyy	
	mm/yy			



Time	:time		:time
Email	:isemail		:isemail
Zip	:zip	country	us – Default US Zip Code format
			uk - United Kingdom Zip Code format
			:zip :zip(country=uk)
Zip + 4	:zip4		:zip4
Phone	:phone	country	us – Default US phone # format
			uk - United Kingdom phone # format
			:phone :phone(country=uk)
Social Security Number	:ssn		:ssn
Using regular expression for validating data in a Text Tag	:custom	regexp	A regular expression expressed in the form of a string. In this syntax, the backslash char that is part of regexp constructs such as "\w", must always be escaped with a backslash, i.e., "\\w". And in order to search for a backslash, one must use \\. The string begin and end character double-quotes ("). If a quote is expected to occur within the data that the user enters, to provide a regexp validation string to match the quote they must be escaped as \" in the expression. A good reference for the syntax of regular expressions used by EchoSign can be found <a href="#">here</a> .
		msg	Message to be displayed to the user when the data does not match the specified expression.
			:custom(regexp="^5[1-5][0-9]{14}\$") :custom(regexp="^.{3}\$",msg="Not a match for three chars in a row!")

A single field can only include one directive for validation. If multiple validation rules are specified in a single Text Tag only the first validation rule gets applied. All subsequent validation rules get ignored.



## Masking Sensitive Data

EchoSign supports masking sensitive data collected during the signature process by adding a specific directive to the Text Tag. During signing, the data entered by the signer in such a field is converted into asterisks (\*) or to a user defined masking character as soon as the signer exits the field. The signer-entered data is also not presented to any subsequent signers or included in the completed PDF. The signer-entered data can be exported from EchoSign through the web UI or through the EchoSign Web Services API.

Text Tag definition:

```
CC Number: {{CC_es_ :signer1:mask }} 
```

During signing:

```
CCNumber: ***** 
```

In the final PDF:

```
CC Number: ***** 
```

<code>{{*CC_es_ :signer1:mask}}</code>	Defines a required field called "CC" assigned to the first signer which gets masked during signing.
<code>{{*CC_es_ :signer1:mask(char=-)}}</code>	Defines a required field called "CC" assigned to the first signer which gets masked during signing with "-". If a character is not specified in the Text Tag, asterisk (*) is used as the default masking character.

## Specifying Conditional Rules For Fields

EchoSign supports applying conditional rules for fields in a document. Conditional rules can be used to define when a particular field within a document is shown to or hidden from the signer during signing. Conditions applied to a particular field are based on the values of one or more other fields within the document. Conditional rules applied to a particular field can also specify that a particular field will be shown or hidden when all the conditions defined for a particular field are satisfied or when any one of the conditions defined for a particular field is satisfied.

<code>_es_ :signer1:showif(condition)</code> <code>_es_ :signer1:showifany(condition)</code>	Show the field to the first signer if any one of the conditions defined is satisfied.
<code>_es_ :signer1:showifall(condition)</code>	Show the field to the first signer only if all the defined conditions are satisfied.
<code>_es_ :signer1:hideif(condition)</code> <code>_es_ :signer1:hideifany(condition)</code>	Hide the field from the first signer if any one of the conditions defined is satisfied.
<code>_es_ :signer1:hideifall(condition)</code>	Hide the field from the first signer only if all the

	defined conditions are satisfied.
<code>_es_:signer1:enableif(condition)</code> <code>_es_:signer1:enableifany(condition)</code>	Enable the field for the first signer if any one of the conditions defined is satisfied. This is similar to the 'showif' condition except that if the condition is not satisfied, the field appearance when during signing will be visible but grayed out and not editable. The grayed out field will display the field's default value, if applicable. The value will also appear in the signed PDF.
<code>_es_:signer1:enableifall(condition)</code>	Enable the field for the first signer only if all the defined conditions are satisfied.
<code>_es_:signer1:disableif(condition)</code> <code>_es_:signer1:disableifany(condition)</code>	Disable the field for the first signer if any one of the conditions defined is satisfied. The disabled field will appear as a grayed out field during signing. If there is a default field value, the value of a disabled field will still appear in the signed PDF.
<code>_es_:signer1:disableifall(condition)</code>	Disable the field for the first signer only if all the defined conditions are satisfied.

A condition is defined in the following way: **fieldname operator value** where the **fieldname** is the name of the form field within the document on which the condition is based. The supported **operators** and valid **values** for the various supported form fields are shown in the tables below.

### Conditional Field Operators

Operator	Meaning	Value	Example
<code>=, ==, eq</code>	Equals	Single value	<code>myCheck=checked</code>
<code>!=, &lt;&gt;, neq</code>	Not Equals	Single value	<code>signature!=signed</code>
<code>&lt;, lt</code>	Less Than	Single value, numeric	<code>myNumber &lt; 10</code>
<code>&gt;, gt</code>	Greater Than	Single value, numeric	<code>myNumber &gt; 40</code>
<code>&lt;=, lte</code>	Less Than or Equals	Single value, numeric	<code>myNumber &lt;= 15</code>
<code>&gt;=, gte</code>	Greater Than or Equals	Single value, numeric	<code>myNumber &gt;= 0</code>
<code>?=, in</code>	In	Comma delimited list, must be quoted or within parentheses	<code>option ?="1,2,3"</code>
<code>?!=, !in, not in, notin</code>	Not In	Comma delimited list, must be quoted or within parentheses	<code>option ?!= "1,2,3"</code>
<code>~=, contains</code>	Contains	Single value	<code>Title ~= Manager</code>
<code>~!=, !contains, not contains, notcontains</code>	Does Not Contain	Single value	<code>Title ~!= Junior</code>

Multiple conditions for a single field can be separated using a comma. For example: `{{notRed_es_.showifall(checkbox=checked,Color!=Red)}}` defines a field called "notRed" which is shown to the signer when both the conditions are satisfied.

### Conditional Field Value For Supported Text Tag Fields

Field Type	Valid Values	Example
Signature	signed, unsigned	signHere=signed
Initials	signed, unsigned	initHere!=signed
Check Box	checked, unchecked	myCheck=checked
Radio Button	Value of the radio button options	Color=Red
Text Fields	Value of the Text Field	Title!=Senior Title~=Chief Price>100

Conditions can be assigned to any field, including required fields like initials, signatures, and signature blocks. When a condition is defined on a required field it will only be treated as required only if the condition makes the field visible to the signer during the signature process. In other words if a signature field has a condition on it that hides it, it will not be visible to the signer and therefore does not require the signer to provide a signature in this particular field. The "required field" count at the bottom of the signing page gets dynamically updated based on the defined conditions to accurately reflect the number of fields which the signer is required to fill before completing signing the document.

Every signer must have one signature field that does not have a condition assigned to it. Because of this, if all signature fields for a given signer have a condition, a new signature block will be added for that signer at the end of the document.

### Specifying Calculation Formulas For Fields

EchoSign supports dynamically populating field values into the document during signing based on other field values/choices made by the customer during signing; these types of fields are called Calculated Fields. Calculation formulas can be defined similar to any other field on the document using Text Tags or through the Web-browser-based Form Field Authoring environment. A calculation formula can reference one or more existing fields in the document and can also be based on another calculated field within the document. Calculations can be applied to numeric fields, date fields and text fields.

In addition to calculating the value of a field in the document, calculated fields can also include formatting information to appropriately style the output of the calculated value in the document. Additional rules such as Conditional Rules, Field Masking, etc. can also be applied to Calculated Fields through Text Tags.

<code>f1_es_.signer1:calc(formula)</code>	A calculated field assigned to the first signer the value for which gets automatically calculated based the defined formula.
<code>f2_es_.signer1:calc(formula):format(formatType, format)</code>	A calculated field the value for which gets calculated based on the defined formula and the resulting output value is of a specific type and is formatted per

	the format specified.
--	-----------------------

Example: A Calculated Field can be used to dynamically calculate the total tax owed on an order.

```

{{Tax_es_:signer1:calc(subtotal * taxrate):format(number, "$,.00")}}

```

The tax calculation is based on the total cost of the items ordered by the customer and the tax rate for the state of residency of the signer. The resulting calculated Tax value will be formatted with a dollar (\$) symbol before the value and will include two decimal point in the resulting value. Say if the subtotal is 1250 and taxrate is 6.25% the resulting calculated value that will be displayed in the Tax field in the document would be \$1,328.13.

For more information about Calculated Fields and using them in your documents please refer to the [Adobe EchoSign Calculated Fields Guide](#).

### Shortening Text Tags

Text Tags provide a very powerful mechanism for defining different types of fields within a document, assigning those fields to specific signers, specifying validation rules on the fields and/or applying conditional rules to those fields. Applying all of this to a single form field can make the field definition very long and can cause the document structure to be changed as a result of the field definition. Furthermore, since each field definition must fit on one line, the use of multiple complex directives can make document layout very difficult. To ensure consistency of the structure of the document, EchoSign supports a mechanism for defining shortened Text Tags within the document. A shortened Text Tag consists of two parts:

- Reference Tag:** A shortened tag definition placed at the appropriate location within the document. The reference tag controls the size and location of the actual form field in the document. A reference tag has the following syntax: `{{$shortened_name}}`. For example: `{{$r}}` defines a reference tag where "r" is the short name of the tag. The size of the field can be adjusted using whitespaces either immediately after the last open curly brace or immediately preceding the first closed curly brace.
- Definition Tag:** This refers to the complete definition for a specific tag and can be defined anywhere in the document. The size and location of the definition tag does not determine the size and location of the actual form field within the document. A definition tag has the following syntax: `{{#shortened_name=Standard Tag Syntax}}`. For example: `{{#r=[ReceiveCommunication_es_:signer1]}}` defines a complete tag definition for the reference tag named "r" – in this case the tag defined a checkbox assigned to the first signer.

For example: Figure below show two Reference Tags placed within a document.

<pre> {{\${a}} I would like to receive e-mail communications. E-mail: {{\${e} </pre>
--



The complete definition of the two Reference Tags is defined elsewhere in the document as shown in the figure below

```

{{#a=(email) TypeOfCommunication_es_:signer1:showif(ReceiveCommunication="checked")}}
{{#e=*emailAddress_es_:signer1:isemail:showif(TypeOfCommunication="email")}}

```

The complete definition for the Text Tags is very long and without the ability to define shortened tags in the appropriate location within the document it would not have been possible to maintain the structure of the original document.

### Removing Pages From Output

By separating form field locations from the definition of the field, it is possible that all the definitions for tags within a document are placed at the end of the document or on a completely separate page within the document. When the document is sent through EchoSign, all the Text Tags are processed and a white box is placed on top of the Text Tag definition so the text is no longer visible to the signer. This may result in entirely blank pages within the document after being processed by EchoSign – pages that contain only Text Tags. To avoid additional unwanted blank pages on the document EchoSign supports a special Text Tag definition to remove a particular page from the processed output document generated by EchoSign before sending the document for signature.

<pre> {{#REMOVE_PAGE_FROM_OUTPUT}} </pre>	<p>Special directive that instructs EchoSign to remove the page on which this directive appears from the processed output for the document. The tag can appear anywhere on the page. A document sent to EchoSign must contain at least one page. If this directive appears on all the pages of a document sent to EchoSign, the first page would not be deleted.</p>
---	--

### Advanced Text Tags Examples

Below are a few examples of Text Tags with description for each tag

#### Check Boxes

- `{{[_es_:signer1]}}` – A check box, assigned to the first signer.
- `{{[*]}}` – A mandatory check box.
- `{{[!]}` – A read-only check box.

#### Radio Buttons

- `{{(Daily)Rate_es_:signer1}}` – A radio button assigned to signer1 with the value of Daily.

#### Drop Down List

- `{{*Color_es_:signer1:dropdown(options="Red,Green,Blue")}}`  
A required drop-down list assigned to the first signer with three possible options to select
- `{{Color_es_:signer1:dropdown(options="Red,Green,Blue"):default("Red")}}`  
A drop-down list assigned to the first signer with three possible options to select and the option "Red" is selected by default. If the value is specified in the definition of the drop-down list, the value must be used to specify the default otherwise the option is used.

- `{{Color_es_:dropdown(options="Red,Green,Blue", values="R,G,B"):default("R"):mask}}`  
A drop-down list not assigned to any specific signer with three possible options to select corresponding to three export values. The value "R" corresponding to the user choice of "Red" is selected by default and the resulting selected option gets masked (displayed as "\*") in the PDF.
- `{{Color_es_:dropdown(options="Red,Green,Blue"):showif(Rate="Daily")}}`  
A drop-down list not assigned to any specific signer with three possible options to select. The drop-down list is only shown if the value of the Radio button called Rate is = "Daily".

### Field Validation

- `{{Birthday_es_:signer2:isdate}}`  
– An optional date field allowing dates in US date formats
- `{{*Zip2_es_:signer2:zip}}`  
– A required field for the second signer to provide a valid Zip code. The field validates the value entered by the signer.

### Shortening Text Tags

- `{{${r}}` – A reference tag specifying a location of a field.
- `{{#r=[Spouse_es_:prefill]}` – A definition for a check box to be prefilled by the sender.

Text tag shortening can also be used as functions and not just to replace an entire field.

For example, let's use the following tag:

```
{{field1_es_:signer:showif(fieldA='Yes')}}
```

This above tag could also be shortened using the following:

```
{{field_es_:${A}.${B}}
```

```
#A=signer
```

```
#B=showif(fieldA='Yes')
```

Text tag shortening also can behave like methods for calculations.

For example, let's use the following tag:

```
{{field1_es_:calc(a * b / c)}}
```

This above tag could also be shortened using the following:

```
Field1_es_:${1(a,b,c)}
```

```
#1(option1, option2, option3) = calc([option1] * [option2] / [option3])
```

### Conditional Rules

- `{{f_es_:hideifall(sig=unsigned,initials=unsigned)}`  
– A field that will show only if sig and initials are both signed.
- `{{*emailAddress_es_:signer1:isemail:showif(TypeOfComm="receiveEmail")}}`

- A required field assigned to the first signer to enter a valid email address. The field is only displayed when another field called "TypeOfComm" has a value of "receiveEmail".
- `{{f_es_:hideifany(Color="Red",Color="Green")}}`
  - A field that will be hidden if the user chooses either the color Red or the color Green in the drop-down list called Color.

### Calculated Fields

- `{{shipState_es_:signer1:calc([billingState]):showif(sameAs=unchecked)}}`
  - A field in the document assigned to the first signer that will only be displayed when another field (a check box) called "sameAs" is unchecked. The value of the field is automatically populated with the value of the field "billingState" that the signer has entered in the document.
- `{{total_es_:signer1:calc(q1*pr1):format(Number,"$,0.00"):align(right)}}`
  - A field in the document assigned to the first signer. Its value gets automatically calculated as the product of the values in fields "q1" and "pr1". The resulting value is formatted as currency and displayed with a dollar (\$) symbol in front of the resulting value and containing two decimal points. The resulting value is also aligned right aligned within the field.
- `{{price_es_:signer:calc(if(pn1="W1",9.99,if(pn1="W2",5.99,if(pn1="W3",24.99,0))))}}`
  - A field in the document called "price" which gets calculated based on the value selected by the signer in another fields "pn1" in the document. In this example if field "pn1" equals W1 the value displayed in the "price" field is 9.99, if field "pn1" equals W2 the value displayed in the "price" field is 5.99 and so on. If none of the values match the value of the field "pn1" the value displayed in "price" is 0. Using this mechanism Calculated Fields can be used to automatically populate the price of an item based on the selection that a user has made during signing.

For more information about calculated fields see [Adobe EchoSign Calculated Fields Guide](#).