



---

Financial & Corporate Compliance

# SmartSign Plus

Tagging Guide

2026.03

---

# Financial & Corporate Compliance

This publication was written for SmartSign Plus®.

## **Publication Information / Version**

Document Title: SmartSign Plus® Tagging Guide

Publication Date: 03 2026

## **Distributed Subject to Terms of a License or other Agreement**

The contents of this publication, including its appendices, exhibits, and other attachments, as updated or revised, are highly confidential and proprietary to Wolters Kluwer Financial Services, Inc. or its subsidiaries or affiliates (“Wolters Kluwer”). This publication is distributed pursuant to a Non-Disclosure Agreement, Evaluation Agreement, License Agreement and/or other similar agreement(s) with Wolters Kluwer or its subsidiary or affiliate. Unless otherwise specifically provided in such agreement(s), the reproduction of this publication is strictly prohibited. Use and distribution of this publication are also subject to the responsibilities and obligations of such agreement(s), which require confidential treatment of this publication and its contents.

Information in this guide is subject to change without notice and does not represent a commitment on the part of Wolters Kluwer.

## **Do Not Reproduce or Transmit**

Unless otherwise specifically authorized in the agreement or license under which this publication has been provided, no part of this publication may be posted, played, transmitted, distributed, copied or reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, or retaining on any information storage and retrieval system, without prior written permission from Wolters Kluwer.

Requests for permission to reproduce content should be directed to Wolters Kluwer, Customer Support by telephone at 1-800-397-2341.

## **Not a Substitute for Legal Advice**

This publication is intended to provide accurate and authoritative information about the subject matter covered based upon information available at the time of publication. Examples given in this publication are for illustrative purposes only.

Development of this publication and the software (including forms, disclosures, reports, and other documents generated by the software) or other products that it describes was based on Wolters Kluwer’s understanding of various laws, regulations and commentaries. Wolters Kluwer cannot and does not guarantee that its understanding is correct.

This publication is not intended, and should not be used, as a substitute for legal, accounting, or other professional advice. Wolters Kluwer is not engaged in providing legal, accounting or other professional services. If legal or other professional assistance is required, you should seek the services of a competent professional. We encourage you to seek the advice of your own attorney concerning all legal issues involving the use of this publication and any products described in this publication. If your interpretations or your counsel’s interpretations are contrary to those expressed in this publication, you should of course, follow your/your counsel’s interpretations.

The following notice is required by law:

Wolters Kluwer’s PRODUCTS AND SERVICES ARE NOT A SUBSTITUTE FOR THE ADVICE OF AN ATTORNEY.

## **Warranty Disclaimer**

Except only for the warranties (if any) expressly set forth in the agreement(s) under which this publication is provided (i.e., your agreement or license for the described product), this publication is provided “as is”, and Wolters Kluwer makes no warranty, express, implied, by description, by sample or otherwise, and in particular and without limitation, makes no implied warranties of merchantability or fitness for purpose. No modifications to this Warranty Disclaimer are authorized unless they are included in a signed, duly executed, written agreement between you and the Wolters Kluwer entity licensing the product described in this publication.

## **Attributions and Acknowledgements**

All trademarks are the property of their respective owners.

## **Copyright Information**

©2026 Wolters Kluwer N.V. and/or its subsidiaries. All rights reserved.

This publication is the confidential information of Wolters Kluwer. Distribution of this publication is subject to restrictions in the license or agreement under which this publication is provided to authorized Wolters Kluwer Financial Institution customers.

---

# Contents

<b>Tagging Introduction.....</b>	<b>4</b>
Tag Types.....	4
<b>Tagging Method Overview.....</b>	<b>5</b>
Manual Tagging.....	5
How to Manually Tag Documents.....	5
XY Coordinates.....	6
XY Coordinate Considerations.....	6
XY Coordinate JSON Sample.....	7
PDF Form Field Tagging.....	8
Field Preparation.....	9
PDF Form Field JSON Data Schema.....	10
PDF Form Field Parameters by Tag Type.....	11
PDF Form Field JSON Sample.....	12
PDF Form Field Using v3 Post Documents.....	13
Adding Documents with Prefilled Tags and PDF Form Fields.....	14
Anchor Text.....	15
Individual tag specs.....	15
Supplemental file specs.....	16
API Updates.....	17
Application Behavior.....	17
<b>API Notes.....</b>	<b>18</b>
Adding Redacted Tags via API.....	19
Updating Tag Type Values via API.....	20
Adding Documents with Prefilled Tags and PDF Form Fields.....	21
<b>Conditional Tagging.....</b>	<b>22</b>

# Tagging Introduction

SmartSign Plus captures participant input in the Signing Room with interactive fields overlain on documents. These fields are referred to as *Tags* within SmartSign Plus.

These tags must be placed carefully in the correct location and assigned to the appropriate participant to effectively capture participant input and reflect the content in a way familiar to stakeholders. This process is referred to as Tagging within SmartSign Plus and closely resembles the paper process of writing content on a physical document with a pen.



**Tip:** New in the 5.7.1 release, you can now update the type value on an existing tag to submit updates that include type ( and type-specific) properties. See [Updating Tag Type Values via API](#) on page 20 for more details.

## Tag Types

SmartSign Plus supports a variety of tag types, each representing different the actions participants are prompted to perform to complete the tag action.

<b>Signature</b>	Places a signature block.
<b>Initials</b>	Places a small signature block for participants to initial.
<b>Text</b>	Places a field used to enter a single line of alphanumeric content.
<b>Multi-line Text</b>	Places a field used to enter multiple lines of alphanumeric content.
<b>Date Picker</b>	Places a pop-up calendar selector.
<b>Check Box</b>	Places a checkbox. Participants may check the box or leave the box clear.
<b>Radio Button</b>	Places a list of yes/no indicators radio buttons. Participants must select only one option from the list.
<b>Multi-Select</b>	Places a list menu of options. Participants may select one or more options from the list.
<b>Drop-Down</b>	Places a menu list of options. Participants may select from the options in the menu.
<b>Auto-Full name</b>	Places a text box that automatically populates the participant's first, middle, and last names (and suffix).
<b>Auto-date</b>	Places a date field that automatically populates with the date on which the document was completed.
<b>Auto-First name</b>	Places a text box that automatically populates the participant's first name.
<b>Auto-Last name</b>	Places a text box that automatically populates the participant's last name.
<b>Auto-Email address</b>	Places a text box that automatically populates the participant's email address.

# Tagging Method Overview

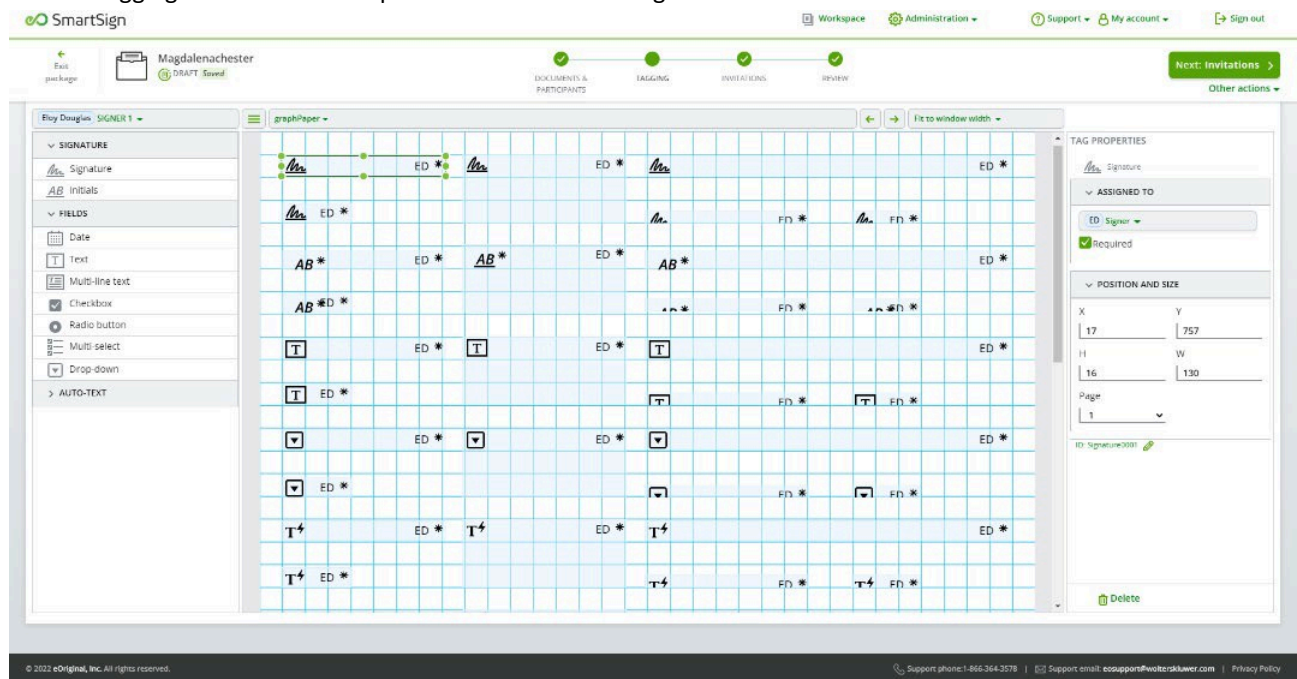
There are several tagging methods available within SmartSign Plus, and the number of available methods continues to increase. Each tagging method has advantages for different client use cases and technical capabilities.

- [Manual Tagging](#)
- [XY Coordinates via API](#)
- [PDF Form Fields via API](#)
- [Anchor Text](#)

## Manual Tagging

Manual tagging is the most technically accessible, but least time-efficient method of tagging. Manual tagging requires manually dragging and dropping individual tags into the document via the back-office interface in SmartSign Plus. This process is very click intensive and is prone to human error.

Manual tagging is effective for simple documents and for organizations with low volume.



## How to Manually Tag Documents

1. On the Tagging page, select the document you want to tag from the drop-down at the top of the editor window.
2. Click the participant drop-down menu in the upper left side and select a participant.
3. Click and drag signature blocks and other data collection fields into the editor window. Position the fields on the page as you want them to display during the signing event.
4. Click a tag to edit properties.
5. Configure data validation for the tag. (Optional)
6. Repeat Steps 1 through 5 to add the for each participant, as needed.
7. Add auto-text to any of the documents in the package. (Optional)
8. Click **Next: Invitations** when tagging is completed.

---

## XY Coordinates

XY Coordinate tagging is a fully automated method of tagging via API. This method relies on the use of location communicated as a set of numbers that translate to a position on the document. The schema for the SmartSign+ XY coordinate tagging is fully documented with samples on our Swagger page.

### SmartSign Plus Swagger Page - POST Tags API

#### [SmartSign Plus Swagger](#)

When applied at runtime (dynamically applied when documents are generated) this tagging method can be a fully comprehensive solution for eSign tagging via API.

If applied as static metadata related to documents ahead of runtime, this is only effective for documents with static content.

## XY Coordinate Considerations

### Origin

The origin is what SmartSign Plus defines as the location representing “0,0” on the coordinate map. SmartSign+ uses the bottom-left corner of the document as the origin.

### Units of Measurement

SmartSign Plus uses PDF points as the unit of measurement on the coordinate map. For example, 1,1 relates to a position that is 1 PDF point above and to the right of the origin.

### Position

#### Page

This is the page of the document where the tag should be placed. Each page has its own 0,0 origin.

#### Left

This is the X coordinate that relates to the bottom-left corner of the tag that is being placed and is relative to the page origin.

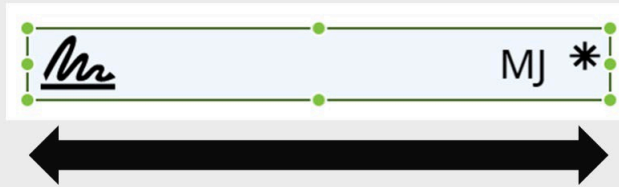
#### Bottom

This is the Y coordinate that relates to the bottom-left corner of the tag that is being placed relative to the page origin.



### Width

This is the number of PDF points that will comprise the tag size from the left to right.



## Height

This is the number of PDF points that will comprise the tag size from the top to bottom



## XY Coordinate JSON Sample

### XY Coordinate JSON Sample

```
{
  "tags": [
    {
      "label": "Tag1",
      "tagName": "Tag1",
      "type": "SIGNATURE",
      "position": {
        "page": 1,
        "left": 38,
        "bottom": 78,
        "width": 132,
        "height": 16
      },
      "participantId": "2MfQMjtDxiawjCtJcVDjJU",
      "role": {
        "type": "Signer",
        "roleNumber": 1
      },
      "required": true,
      "appearanceStyles": {
        "font": {
          "type": "Helvetica",
          "style": "Bold",
          "size": 12,
          "color": "#000000"
        }
      }
    }
  ]
}
```

```
}  
]  
}
```

## PDF Form Field Tagging

SmartSign Plus supports the direct import of documents with PDF Form Field tags to generate native SmartSign Plus tags. A proprietary schema must be inserted into the value of the Name field in the PDF form field (Figure 1). SmartSign Plus extracts the coordinates directly from the form field properties and uses the schema in the Name value to generate eSign tags (Figure 2.)

The screenshot shows a PDF document titled "PDFFX Expert 01.pdf" in the PDF Expert application. The document contains the following text:

Tampa, FL 33602  
800-555-2000

Please return to Lender at the address listed in the Date and Parties section.

**Signatures**

**Signatures for Owner(s) and Authorization to Insurance Agent and Company.** By signing below, I agree to the terms contained in this Agreement and acknowledge receipt of a copy of this Agreement. I request the listed Insurance Company and Agency to provide the indicated coverage and list you on the policy with the indicated status. I also request the Insurance Company or its authorized Agent to immediately confirm that the policy is in effect by signing this form and forwarding a copy of the policy to you.

**Owner**

There are two signature blocks, each with a signature field and a date field. The signature fields contain JSON tags:

Left block (Joseph J Doe):  
Signature field: `{ "name": "Signature001", "type": "SIGNATURE", "role": "type": "S", "tagname": "Date001", "type": "AU"`  
Date field: `{ "name": "Date001", "type": "DATE", "role": "type": "D", "tagname": "Date001", "type": "AU"`

Right block (Josephine K Doe):  
Signature field: `{ "name": "Signature002", "type": "SIGNATURE", "role": "type": "S", "tagname": "Date002", "type": "AU"`  
Date field: `{ "name": "Date002", "type": "DATE", "role": "type": "D", "tagname": "Date002", "type": "AU"`

At the bottom of the page, there is a footer with the following text:

Agreement to Provide Insurance  
© 2018 Wolters Kluwer Financial Services, Inc.  
All rights reserved.

M2\_IDTran03\_FL\_Freddie\_FixedRate  
2022011421.2.0.4024-J0211026N

05/2021  
Page 2 of 3

Figure 1: Figure 1



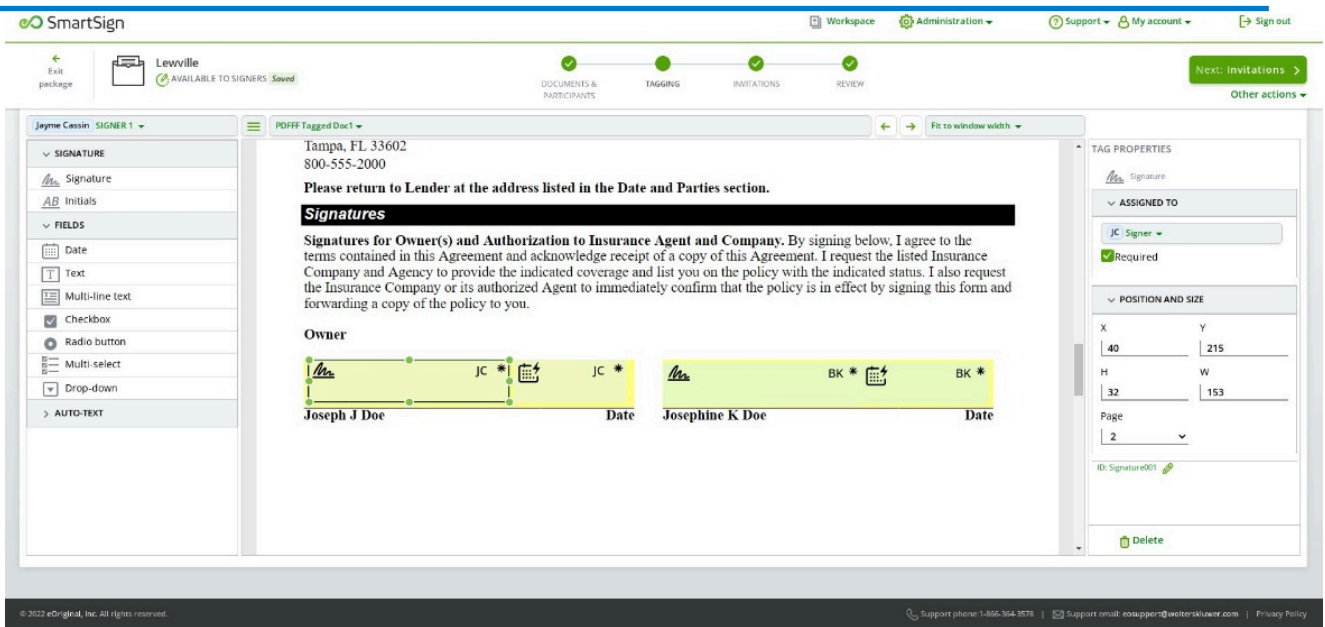


Figure 2: Figure 2

## Field Preparation

Form fields in the PDF should be placed at the location in which the preparer intends for interaction to occur.

The type of the PDF Form Field Object (signature, date, text, etc.) is ignored. Tag location is pulled directly from the PDF Form Field properties; no additional input is required by the preparer.

JSON data following the SmartSign Plus tagging schema (found in the [Data Schema](#) section of this document) is inserted in the Name value of the PDF Form Field (Figure 3) and defines all critical data including:

- Tag type
- Assigned participant
- Unique tag name
- Required/optional
- Informational label
- Dropdown selections
- Tag styling
- Validation

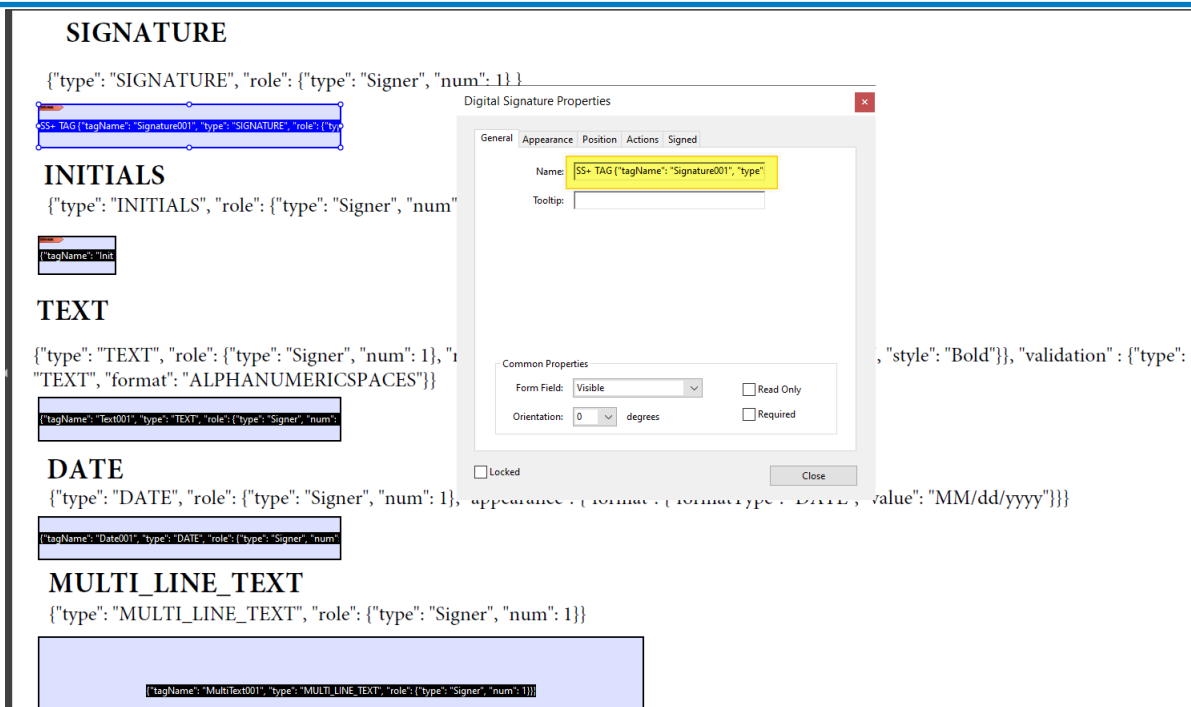


Figure 3: Figure 3

## PDF Form Field JSON Data Schema

Name	Details								
label	string title: name.v1 example: My Example's Name minLength: 1 maxLength: 100								
tagName	string title: name.v1 example: My Example's Name minLength: 1 maxLength: 100								
type*	string Enum: [ TEXT, DATE, RADIO_BUTTON, MULTI_SELECT, INITIALS, SIGNATURE, MULTI_LINE_TEXT, CHECKBOX, DROPDOWN, FULL_NAME, AUTO_DATE, FIRST_NAME, LAST_NAME, EMAIL ]								
role	<table> <tr> <td>type*</td><td>string</td></tr> <tr> <td>roleNumber*</td><td>integer</td></tr> <tr> <td></td><td>minimum: 1</td></tr> <tr> <td></td><td>maximum: 9</td></tr> </table>	type*	string	roleNumber*	integer		minimum: 1		maximum: 9
type*	string								
roleNumber*	integer								
	minimum: 1								
	maximum: 9								

validationRule	min	number
	max	number
	type*	string title: tag-validation-type.v1Enum:
	format	string title: tag-validation-format.v1Enum:
	excludes	string title: tag-date-exclusion.v1Enum:
	minDate	string(\$date)
	maxDate	string(\$date)
	numDays	integer minimum: 0
selectOptions	string	
appearanceStyle s	font {	
		type string style string size integer color string
	}	
	format {	
		formatType String Enum: [ DATE, NUMBER ] value string }
participantId*	string(\$uuid)	
required	boolean default: true	

## PDF Form Field Parameters by Tag Type

Parameter	Tag Type(s)
label	Allowed for all tag types.
tagName	Allowed for all tag types.
type*	Required for all tag types.

Role*	Required for all tag types.
validationRule	Allowed for TEXT, DATE, & MULTI_LINE_TEXT; Not used for other tag types
selectOptions	Required for DROPDOWN & MULTI_SELECT; Not used for other tag types
appearanceStyles	Allowed for TEXT, DATE, MULTI_LINE_TEXT, & AUTO_DATE; Not used for other tag types
participantId*	Required for all tag types.
required	Allowed for all tag types.

## PDF Form Field JSON Sample

Option1: If the field is a TEXT

```
{
  "executionType": "ESign",
  "name": "PDF Tagged Doc1",
  "order": 1,
  "parseFormFieldsFromDocument": true,
  "participants": [
    {
      "participantUid": "{{ParticipantID}}",
      "state": "REQUIRED"
    }
  ],
  "tags": [
    {
      "label": "My Example's Name",
      "tagName": "My Example's Name",
      "type": "TEXT",
      "role": { "type": "SIGNER", "roleNumber": 1 },
      "validationRule": {
        "type": "TEXT",
        "format": "ALPHANUMERIC",
        "min": 0,
        "max": 0
      },
      "appearanceStyles": {
        "font": { "type": "Helvetica", "style": "Italic", "size": 10, "color": "#000000" }
      },
      "participantUid": "{{ParticipantID}}",
      "required": true
    }
  ]
}
```

Option 2: If the field is a DATE

```
{
  "executionType": "ESign",
  "name": "PDF Tagged Doc1",
  "order": 1,
  "parseFormFieldsFromDocument": true,
  "participants": [
    {
      "participantUid": "{{ParticipantID}}",
      "state": "REQUIRED"
    }
  ]
}
```

```

    ],
    "tags": [
      {
        "label": "My Example's Name",
        "tagName": "My Example's Name",
        "type": "DATE",
        "role": { "type": "SIGNER", "roleNumber": 1 },
        "validationRule": {
          "type": "DATE",
          "minDate": "2022-03-25",
          "maxDate": "2022-03-25",
          "numDays": 0,
          "excludes": "WEEKENDS"
        },
        "appearanceStyles": {
          "font": { "type": "Helvetica", "style": "Italic", "size": 10, "color": "#000000" },
          "format": { "formatType": "DATE", "value": "MM/dd/yyyy" }
        },
        "participantUid": "{{ParticipantID}}",
        "required": true
      }
    ]
  }
}

```

## PDF Form Field Using v3 Post Documents

### Adding SmartSign Plus Enhanced Form Fields

```

{
  "executionType": "E-Sign",
  "name": "Doc Old",
  "order": 1,
  "participants": [
    {
      "participantUid": "{{ParticipantID1}}",
      "state": "REQUIRED"
    }
  ],
  "parseFormFieldsFromDocument": ["SSPLUS_ENHANCED_FORM_FIELDS"]
}

```

### Adding Non-SmartSign Plus Enabled Form Fields

The POST Document API updated from v2 to v3 supports the ability to read and add non-SmartSign Plus enabled PDF form field properties. This functionality allows users to retrieve detailed information about form fields within a document, making PDF form handling more efficient and consistent.

Specifically, the optional parameter: *parseFormFieldsFromDocument*, allows the following options:

- **SSPLUS\_ENHANCED\_FORM\_FIELDS** – Adds only SmartSign Plus-enabled form fields (same behavior as the v2 POST Documents call).
- **PDF\_FORM\_FIELDS** – Adds only non-SmartSign Plus-enabled PDF form fields.
- **NULL** – Uploads the document without parsing any form fields.

These parameters must be passed in the request body to ensure proper processing.

### Key Details

**PDF\_FORM\_FIELDS:** Adds only non-SmartSign Plus form fields, and tags are unassigned by default. If unassigned fields exist, users cannot finalize the package, triggering a showstopper warning (only for unassigned signature tags).

- **SSPLUS\_ENHANCED\_FORM\_FIELDS**: Adds only SmartSign Plus form fields, with the same behavior as v2. Users can still use the v2 POST Documents call for SmartSign Plus form fields.
- **Both SSPLUS\_ENHANCED\_FORM\_FIELDS and PDF\_FORM\_FIELDS**: You can include both options in the request body to add both types of fields, but the document must contain both SmartSign Plus and non-SmartSign Plus form fields.

```
{
  "executionType": "E-Sign",
  "name": "old new ffl111",
  "order": 1,
  "participants": [
    {
      "participantUid": "{{ParticipantID1}}",
      "state": "REQUIRED"
    }
  ],
  "parseFormFieldsFromDocument": [
    "PDF_FORM_FIELDS"
  ]
}
```



**Note:** Both the SSPLUS\_ENHANCED\_FORM\_FIELDS and PDF\_FORM\_FIELDS values can be used.

Refer to Swagger for additional documentation.

#### Additional Important Considerations

- **parseFormFieldsFromDocument** is an array or null (not the string "NULL"). If parseFormFieldsFromDocument is omitted, the behavior equals null (no parsing).
- **Unassigned Tags**: All tags added with PDF\_FORM\_FIELDS are unassigned until assigned to a participant. If signature tags are unassigned, a showstopper warning is triggered, but other tags can remain unassigned without issue.
- **GET Tags & Deletion Support**: The existing GET Tags API is unchanged. We support deleting non-SmartSign Plus form fields via the Delete Tags request.
- **NULL Behavior**: When NULL is used, the document uploads without parsing any form fields, even if both SmartSign Plus and non-SmartSign Plus fields are present.
- **Field Display**: If a document contains both SmartSign Plus and non-SmartSign Plus form fields but the request only specifies non-SmartSign Plus fields, only those fields display, and SmartSign Plus fields do not appear.
- **Initial Fields**: Initials are allowed as manual tags; they are not available via PDF form-field parsing.
- **Date Field Standardization**: Non-SmartSign Plus Date fields currently support a variety of date formats, and date fields in those formats are now converted to the standard date format supported by SmartSign Plus. Note that standardization applies only to new non-SmartSign Plus form fields.

## Adding Documents with Prefilled Tags and PDF Form Fields

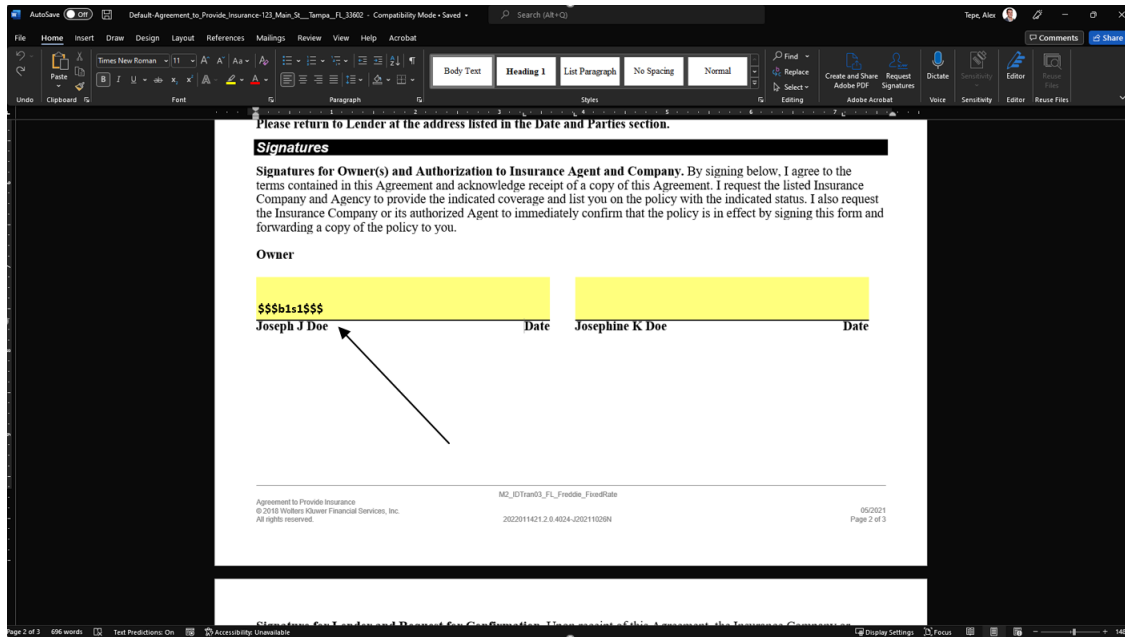
SmartSign Plus supports adding documents that contain prefilled tags and PDF form fields. When these types of documents are uploaded, SmartSign Plus can use the V2 or V3 Tags API with the appropriate PDF form fields parsing parameter. If the document is valid and free of errors, SmartSign Plus will successfully parse and display the pre-filled tags.

```
{
  "executionType": "E-Sign",
  "name": "Doc 3",
  "order": 1,
  "participants": [
    {
      "participantUid": "{{ParticipantID1}}",
      "state": "REQUIRED"
    }
  ],
}
```

```
{
  "participantUid": "{{ParticipantID2}}",
  "state": "REQUIRED"
},
"parseFormFieldsFromDocument": [ "SSPLUS_ENHANCED_FORM_FIELDS" ]
}
```

## Anchor Text

Anchor text provides the ability to embed unique keys in the text of documents parsed by SmartSign Plus to be used as anchors for applying tags to the document as an alternative to X/Y tags or PDF Form fields.



The anchors need to be accompanied by a key. The key in SSWEB is a supplemental file sent with the document and is the v1 solution for our anchor tags here. An additional level to this will be implemented later to include a page in organization settings to map keys to default preferences.



**Important:** We cannot expect that a full set of JSON instructions can be embedded because the text length needs to be minimal to avoid affecting the format of the document.

[Individual tag specs](#) on page 15

[Supplemental file specs](#) on page 16

[API Updates](#) on page 17

[Application Behavior](#) on page 17

## Individual tag specs

Must be unique text within a single document when considering tag properties.



**Note:** The anchor text can be used multiple times in a document if all tag properties (forgiving page number, left, and bottom values) are consistent.

1. Tag ID (optional)
  - a. if not provided, apply ID automatically
2. Tag Type

- 
- a. TEXT, DATE, RADIO\_BUTTON, MULTI\_SELECT, INITIALS, SIGNATURE, MULTI\_LINE\_TEXT, CHECKBOX, DROPDOWN, FULL\_NAME, AUTO\_DATE, FIRST\_NAME, LAST\_NAME, EMAIL
  - 3. Page number
    - a. **applied dynamically**
  - 4. Left
    - a. **applied dynamically**
  - 5. Bottom
    - a. **applied dynamically**
  - 6. Height
  - 7. Width
  - 8. Validation
    - a. Existing schema from POST tag
  - 9. Appearance
    - a. Type Font
    - b. Style
      - 1. Bold, Italic, Outline
  - 10. Participant ID
    - a. Required (yes/no)

## Supplemental file specs

Should closely resemble the POST Tags schema.

Tag location is applied dynamically, but the remaining data must be provided explicitly.

Need to additionally include X/Y offset to determine a point in reference to the bottom left coordinate of the text where the origin of the tag will be placed.

```
{
  "executionType": "E-Sign",
  "name": "Sample Document",
  "order": 1,
  "parseFormFieldsFromDocument": false,
  "participants": [
    {
      "participantUid": "{{ParticipantID1}}",
      "state": "REQUIRED"
    },
    {
      "participantUid": "{{ParticipantID2}}",
      "state": "REQUIRED"
    }
  ],
  "tags": [
    {
      "label": "Signature Field",
      "tagName": "Signature1",
      "type": "SIGNATURE",
      "position": {
        "anchor": "$signature_anchor$",
        "offset": { "x": 0, "y": 0 },
        "width": 120,
        "height": 20
      },
      "participant": {
```



```

      "id": "{{ParticipantID1}}"
    },
    "required": true
  },
  {
    "label": "Signature Date",
    "tagName": "AutoDate1",
    "type": "AUTO_DATE",
    "position": {
      "anchor": "$date_anchor$",
      "offset": { "x": 0, "y": 0 },
      "width": 120,
      "height": 20
    },
    "appearanceStyles": {
      "format": {
        "formatType": "DATE",
        "value": "MM/dd/yyyy"
      }
    },
    "participant": {
      "id": "{{ParticipantID2}}"
    },
    "required": true
  }
]
}

```

## API Updates

Document Definition is updated to allow “contains anchor tags” declaration and should support the format of the anchor tag supplemental file as part of the multi-part form data



**Note:** When a customer submits a doc with anchor tags, they need to let us know we should look for them

Definition	Path	Request	Response
Document	POST  /organizations/{orgId}/ packages/ {packageId}/ executable-documents	Add a new parameter to this call  Tag JSON  <i>see sample</i>	1. HTTP Response 2. Document ID 3. Document Name
Document	PATCH  /organizations/{orgId}/ packages/ {packageId}/ executable-documents	Add a new parameter to this call  Tag JSON  <i>see sample</i>	1. HTTP Response 2. Document ID 3. Document Name
Document	GET  /organizations/{orgId}/ packages/ {packageId}/ executable-documents	Path	No Anchor data response needed

## Application Behavior

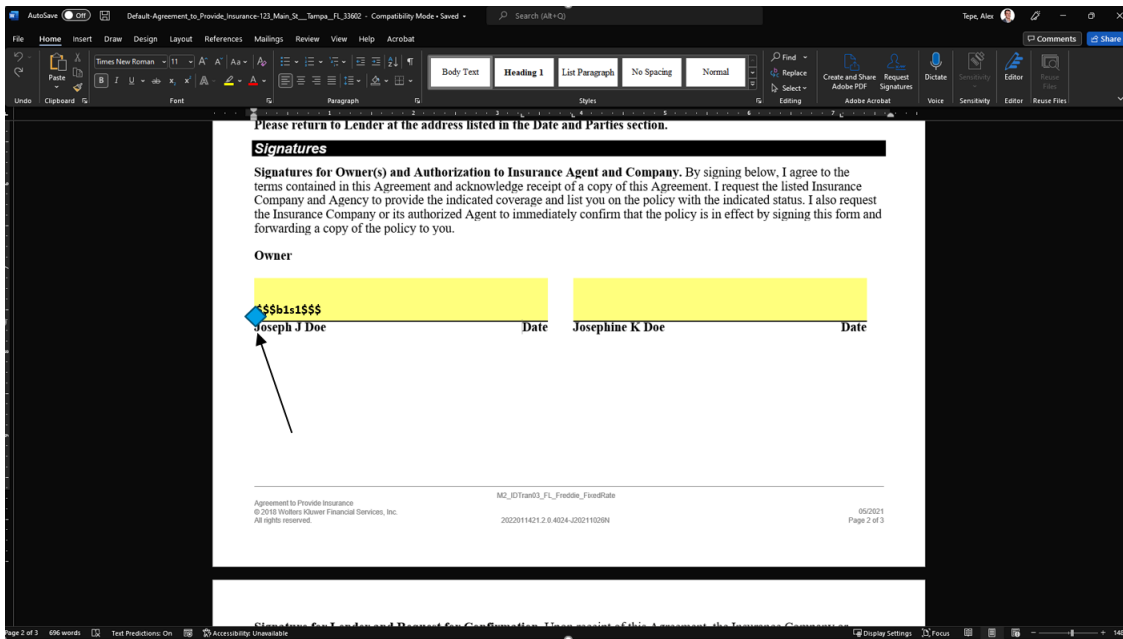
Parse the document, using the bottom left coordinate of anchor text as the tag origin.



**Note:** Tag location is applied dynamically, but the remaining data must be provided explicitly.

When a document contains an anchor tag and is accompanied by the required tag data, then SmartSign Plus should:

1. Parse the document (SSWEB uses iText to do this) to find the anchor locations and content.
2. Map the anchors to the tag properties file.
3. Use the bottom left coordinate of the anchor to determine the tag origin.
4. Construct tags in place using the tag properties associated with the anchor.



In the example above, the blue diamond is at the bottom left coordinate of the anchor text and would be used to dictate the values for:

1. bottom
2. left
3. page

## API Notes

### Document API

The Document API has been updated (beginning with v2 Document API) to support a new parameter *parseFormFieldsFromDocument* in the Document Properties object. By default, this value is set to false. When this value is set to true, SmartSign Plus searches the PDF for form fields and then imports them as tags.

### SmartSign+ Swagger Page (Stage Environment)

- [Adding Redacted Tags via API](#) on page 19
- [Updating Tag Type Values via API](#) on page 20
- [Adding Documents with Prefilled Tags and PDF Form Fields](#) on page 14

## Adding Redacted Tags via API

SmartSign Plus supports the ability to obscure unsigned signature and initial tags and redact sensitive data in text fields within documents. Watermark templates and watermark rules in Admin Settings control when the redaction occurs.

The optional *confidential* parameter is available in POST and PATCH tag APIs and accepts true/false Boolean values.

```
{
  "tags": [
    {
      "label": "SIGNATURE_REQUIRED",
      "type": "SIGNATURE",
      "position": { "page": 1, "left": 50, "bottom": 440, "width": 120,
"height": 16 },
      "participant": { "id": "{{ParticipantID}}" },
      "required": true
    },
    {
      "label": "INITIALS_REQUIRED",
      "type": "INITIALS",
      "position": { "page": 1, "left": 50, "bottom": 400, "width": 30,
"height": 16 },
      "participant": { "id": "{{ParticipantID}}" },
      "required": true
    },
    {
      "label": "TEXT_REDACTED",
      "type": "TEXT",
      "position": { "page": 1, "left": 50, "bottom": 360, "width": 130,
"height": 16 },
      "validationRule": { "min": 1, "max": 30, "type": "TEXT", "format":
"STRINGLENGTH" },
      "appearanceStyles": { "font": { "type": "Times Roman" } },
      "participant": { "id": "{{ParticipantID}}" },
      "required": true,
      "confidential": true
    },
    {
      "label": "MULTILINE_TEXT_REDACTED",
      "type": "MULTI_LINE_TEXT",
      "position": { "page": 1, "left": 50, "bottom": 320, "width": 130,
"height": 16 },
      "validationRule": { "min": 1, "max": 999, "type": "TEXT", "format":
"STRINGLENGTH" },
      "participant": { "id": "{{ParticipantID}}" },
      "required": true,
      "confidential": true
    }
  ]
}
```



### Note:

You can mark a tag as **confidential** to ensure its value is redacted in documents upon signing. This can **only** be done for the following types:

- **Text** and **Multiline Text** fields
- **Unsigned Signatures** (SIGNATURE)
- **Unsigned Initials** (INITIALS)
- Use the **confidential: true** property in your POST /v2/tags request to enable redaction.



**Note:** Please note that in order to support the Data Redaction functionality, SmartSign Plus tags are opaque. If a tag is placed in a way that overlaps any text within the document, signers will be unable to see the text behind the tag.

## Updating Tag Type Values via API

SmartSign Plus supports the ability to update an existing tag's type value. This update enables partial updates that include type (and type-specific properties), while leaving unspecified fields unchanged. For PDF-mapped tags, type changes are allowed only when compatible with the underlying PDF form field; incompatible changes are rejected.

The API is supported when a package is in one of the following draft states.

- DRAFT
- DRAFT\_REACTIVATED
- DRAFT\_ACTIVE\_COLLABORATION



**Note:** If the API is executed when a package is in any other status, an error message is returned.

This feature is available in POST and **PATCH TAG (v3)** API.

```
{
  "tags": [
    {
      "id": "2GY3Xv0IopeHiYx7rrHlWu",
      "type": "SIGNATURE",
      "tagName": "SIGNATURE1",
      "position": {
        "page": 1,
        "left": 81,
        "bottom": 818,
        "width": 89,
        "height": 14
      },
      "required": true,
      "confidential": false,
      "participant": {
        "id": "{{ParticipantID1}}",
        "role": {
          "type": "SIGNER",
          "num": 1
        }
      }
    },
    {
      "id": "KtUB6hU4noZVhJoLbbeKM",
      "type": "INITIALS",
      "tagName": "INITIALS",
      "position": {
        "page": 1,
        "left": 83,
        "bottom": 783,
        "width": 87,
        "height": 15
      },
      "required": true,
      "confidential": false,
      "participant": {
        "id": "{{ParticipantID1}}",
        "role": {
          "type": "SIGNER",
          "num": 1
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "id": "452pAzoWUzuLSxvMsMs8vU",
    "type": "DATE",
    "tagName": "DATE",
    "position": {
      "page": 1,
      "left": 83,
      "bottom": 748,
      "width": 86,
      "height": 23
    },
    "required": true,
    "confidential": false,
    "participant": {
      "id": "{{ParticipantID1}}",
      "role": {
        "type": "SIGNER",
        "num": 1
      }
    },
    "appearanceStyles": {
      "format": {
        "formatType": "DATE",
        "value": "MM/dd/yyyy"
      }
    }
  }
]
}

```

## Adding Documents with Prefilled Tags and PDF Form Fields

SmartSign Plus supports adding documents that contain prefilled tags and PDF form fields. When these types of documents are uploaded, SmartSign Plus can use the V2 or V3 Tags API with the appropriate PDF form fields parsing parameter. If the document is valid and free of errors, SmartSign Plus will successfully parse and display the pre-filled tags.

```

{
  "executionType": "E-Sign",
  "name": "Doc 3",
  "order": 1,
  "participants": [
    {
      "participantUid": "{{ParticipantID1}}",
      "state": "REQUIRED"
    },
    {
      "participantUid": "{{ParticipantID2}}",
      "state": "REQUIRED"
    }
  ],
  "parseFormFieldsFromDocument": [ "SSPLUS_ENHANCED_FORM_FIELDS" ]
}

```

# Conditional Tagging

SmartSign Plus supports conditional tagging, which provides the ability to display specific tags dependent upon the input for other tags and ensures that tags display only when the established conditions have been met. For example, you may choose to make an Initial tag conditional upon a Signature tag so that the Initials field displays only once a Signature has been entered on the document.



## Note:

- Conditional tags depend on the state of a parent tag (e.g., blank, filled, checked, etc.).
- These tags are only rendered or required if the defined condition is met.
- Conditional tagging is not available for Auto Text tags.
- Conditional tagging is supported in Document Templates.

## Setup Instructions

1. Add the parent tag(s) first in an initial POST /v2/tags request.
2. In a separate POST /v2/tags call, add the child/conditional tags and define their dependency using the tagConditionalRules field.

## Example Conditions

Use clear condition names in text (no spaces):

- IS\_BLANK: Checks if the tag's value is blank.
- IS\_EQUAL\_TO: Checks if the tag's value is equal to a specified value.
- IS\_NOT\_BLANK: Checks if the tag's value is not blank.
- IS\_NOT\_EQUAL\_TO: Checks if the tag's value is not equal to a specified value.

## Option 1

```
{
  "tags": [
    {
      "label": "SIGNATURE",
      "type": "SIGNATURE",
      "position": { "page": 1, "left": 50, "bottom": 440, "width": 120, "height": 16 },
      "participantUid": "{{ParticipantID}}",
      "required": true
    },
    {
      "label": "INITIALS",
      "type": "INITIALS",
      "position": { "page": 1, "left": 50, "bottom": 400, "width": 30, "height": 16 },
      "participantUid": "{{ParticipantID}}",
      "required": true,
      "tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition": "IS_NOT_BLANK" }],
    },
    {
      "label": "DATE",
      "type": "DATE",
      "position": { "page": 1, "left": 50, "bottom": 370, "width": 130, "height": 16 },
      "appearanceStyles": { "format": { "formatType": "DATE", "value": "MM/dd/yyyy" } },
    }
  ]
}
```

```

"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "TEXT",
"type": "TEXT",
"position": { "page": 1, "left": 50, "bottom": 340, "width": 130, "height":
16 },
"validationRule": { "min": 1, "max": 30, "type": "TEXT", "format":
"STRINGLENGTH" },
"appearanceStyles": { "font": { "type": "Times Roman" } },
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "MULTI_LINE_TEXT",
"type": "MULTI_LINE_TEXT",
"position": { "page": 1, "left": 50, "bottom": 310, "width": 130, "height":
16 },
"validationRule": { "min": 1, "max": 999, "type": "TEXT", "format":
"STRINGLENGTH" },
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "CHECKBOX",
"type": "CHECKBOX",
"position": { "page": 1, "left": 50, "bottom": 290, "width": 20, "height":
16 },
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "RADIO_BUTTON",
"type": "RADIO_BUTTON",
"position": { "page": 1, "left": 50, "bottom": 260, "width": 12, "height":
12 },
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "DROPDOWN",
"type": "DROPDOWN",
"position": { "page": 1, "left": 50, "bottom": 230, "width": 130, "height":
16 },
"selectOptions": ["Value 1", "Value 2", "Value 3", "Value 4", "Value 5"],
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
},
{
"label": "MULTI_SELECT",
"type": "MULTI_SELECT",

```

```

"position": { "page": 1, "left": 50, "bottom": 200, "width": 120, "height":
16 },
"selectOptions": ["Value 1", "Value 2", "Value 3", "Value 4", "Value 5"],
"participantUid": "{{ParticipantID}}",
"required": true,
"tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
}
]
}

```

#### Option 2 (Simpler Example):

```

{
  "tags": [
    {
      "label": "SIGNATURE",
      "type": "SIGNATURE",
      "position": { "page": 1, "left": 50, "bottom": 440, "width": 120, "height":
16 },
      "participantUid": "{{ParticipantID}}",
      "required": true
    },
    {
      "label": "TEXT",
      "type": "TEXT",
      "position": { "page": 1, "left": 50, "bottom": 340, "width": 130, "height":
16 },
      "validationRule": { "min": 1, "max": 30, "type": "TEXT", "format":
"STRINGLENGTH" },
      "appearanceStyles": { "font": { "type": "Times Roman" } },
      "participantUid": "{{ParticipantID}}",
      "required": true,
      "tagConditionalRules": [{ "parentTagUid": "{{SIGN}}", "condition":
"IS_NOT_BLANK" }]
    }
  ]
}

```



# Wolters Kluwer

## About Wolters Kluwer

Wolters Kluwer is a global provider of professional information, software solutions, and services for clinicians, accountants, lawyers, and professionals in the tax, finance, audit, risk, compliance, and regulatory sectors. We help our customers make critical decisions every day by providing *expert solutions* that combine deep domain knowledge with specialized technology and services. The company is headquartered in Alphen aan den Rijn, the Netherlands.

## About Financial & Corporate Compliance

FCC provides expert compliance services and software solutions for financial institutions, corporations, small and midsize businesses, and law firms with solutions that enable compliance with ever-changing regulatory and legal obligations, improve efficiency, and help achieve better business outcomes. The division offers technology-enabled expert services and software solutions focused on loan compliance, regulatory compliance, legal entity management, and corporate services.

For more information, visit <https://www.wolterskluwer.com>, follow us on [LinkedIn](#), [X](#), [Facebook](#), and [YouTube](#).

---

**Customer Service:** (800) 552-9410 *Available Monday through Friday, 8:00 a.m. to 7:00 p.m., Eastern time.*

**Compliance Solutions Support Portal:** <https://wolters-kluwer.my.site.com/ComplianceSolutionsSupport/s/>

**SmartSign Plus SupportLine:**

**Phone:** 800-274-2711 ext. 1123666

*Available Monday through Friday, 8 a.m. to 8 p.m., Eastern time.*

**Email:** [emailaddress@wolterskluwer.com](mailto:emailaddress@wolterskluwer.com)