SIGNiX Interface Guide: Editable Fields Specification

Version 1.6

October 2019

Introduction

This document specifies support for user-editing of document data.

Legal Considerations

The portion of a document covered by a signature must not change after the signature is created.

Signature coverage is legally determined by what a reasonable person would expect from looking at the visual representation of the document. Since computers are not yet sophisticated enough to reliably determine this, it is necessary to explicitly inform the signing process of the fields covered by each signature, so that the values of these fields can be frozen when the signature is created.

Each signable read/write document field must be listed in the transaction metadata, along with a specification of which signature lines cover it.

Workflow

This section describes the workflow for the server-credential signing method.

A client starts the process by uploading a set of documents and the corresponding transaction metadata.

When a signing line is activated in the workflow, its editable fields will be presented for editing in the document.

Validations can be defined for editable document fields. Validations will be performed when the "Sign" button is pressed. If any errors are found, the signing ceremony will not be performed. The user must make corrections and then push the "Sign" button again.

All fields covered by the signing line will be read only after signing, so that later signers will not be able to edit them.

When all the signatures are complete, the client downloads the signed documents containing the updated field values. The download may optionally include a separate report of field values and the transaction audit trail (see "Field Data Reports" section below).

Document Preparation

There must be fields in each document for all data that is editable and listed in the SubmitDocumentReq XML. Each field and Radio Button group must have a name that is unique within its document (for example, you cannot have two fields named 'Field 1' listed in the same PDF, unless they are Radio Buttons in the same Radio Button group). Fields must also meet any restrictions noted in the Supported Field Types section, below.

Fields that will be updated by some transaction participant must be editable (not read only). Fields that carry tagged data but that are not editable by any signer should be marked read only.

The document must also conform to all the requirements listed under Important Notes in the next section.

Assigning Editing Rights

Field editing rights are assigned to transaction parties in the SubmitDocument request (see /B2BSpec/), as shown in the XML Request Format section below.

The assignment is done using the following elements: <FormStructure> inside <Form>, and <Covers section="*SectionName*"> inside <SignatureLine> and <InitialLine>.

Important Notes

A Transaction can have one or more Forms but each Form must adhere to the following Rules when it comes to Editable Fields:

1. Each <FormField> element must have a name attribute with a unique value.
2. Each <FormField> name attribute must be unique with respect to all types of fields, including <SignField>, <InitialField>, and <DateField> - there must be no duplicate names between these fields.
3. Each <FormField>, <SignField>, <InitialField>, and <DateField> listed in the XML must have a field in the corresponding PDF with a matching name.
4. PDF fields that will be edited or signed (those whose names appear in a <FormField>, <SignField>, <InitialField> or <DateField> element in the XML) must be visible and modifiable in the PDF itself, i.e. they must not have "read only" or "invisible" attributes in the PDF.
5. Any fields that need to be included in the Field Data Report of the DownloadDocument response, but that don't need to be edited by a Signer, should be placed in the "anonymous section" of the <FormStructure>, above the first <Section name>.
6. Each <Section> in a <FormStructure> must have a name attribute with a name that is unique among all the <Section> elements.

7. Only one signature or initial can cover a section. This means that only one <Covers> element, contained in one<SignatureLine> or one <InitialLine>, can have a section attribute whose value matches a given <Section> name. *Best Practice: Associate the section to a signer’s last signature or initial block (allows the use of feature sets).*
8. A <Section> can not be covered by more than one Signer. *Best Practice: Associate to the First Signer (since that's usually the main Client Signer).*
9. Different <Section> elements can be covered by different Signers, and the appropriate Signer will be prompted to fill in the data for that section (and only the data for that section) when it's their turn to Sign. *Best Practice: If Section1 contains fields to be filled out by Signer 1, associate to Signer 1 only. If Section2 contains fields to be filled out by for Signer 2, associate to Signer 2 only.*
10. Elements under a <FormStructure> – whether editable fields, check boxes, or radio buttons – must be listed in the order of appearance on the Form. Having a SubmitDocument request with a radio button grouping as the very last set of fields, while this group actually sits on Page 1 above all other fields, is going to cause viewing and display issues during the signing process and, in some cases, may not work at all.

XML Request Format

```
<?xml version="1.0" ?>
<SubmitDocumentRq xmlns="urn:com:signix:schema:sdddc-1-1">

    ...

    <Data> <!--occurs once only -->

        ...

        <!-- Identifies a party to the transaction, someone who signs or
views the documents -->
        <!-- The order of the MemberInfo   elements affects  the workflow
sequence. -->
        <!-- occurs once for each party -->
        <MemberInfo> <!-- Party P01 -->
            <SSN>000456789</SSN>
            <DOB>10/24/1956</DOB>
            <FirstName>John</FirstName>
            <MiddleName>Q</MiddleName>
            <LastName>Smith</LastName>
            <Email>demo@signix.test</Email>
            <HomePhone>000-456-7890</HomePhone>
            ...
        </MemberInfo  >

        <!-- Additional MemberInfo elements, if needed, appear here. -->

        <!-- Describes one of the documents in the set of documents for this
transaction. -->
        <Form> <!--occurs once for each document -->
            <RefID>SamplesForm2</RefID>
            <Desc>Test Application: Samples</Desc>
            <FileName>Samples_Sample_Application.pdf</FileName>
            <MimeType>application/pdf</MimeType>
            <FormPassword>s4x59*%lks</FormPassword> <!-- Owner password for
the document,
                                                         if any -->

            <!-- Indicates document fields that will be updated and covered
by signatures. -->
            <FormStructure>
                <!-- FormFields outside a Section element are in the
anonymous section. -->
                <!-- The anonymous section is useful for setting attributes
on fields -->
                <!-- that are not editable. -->

                <!-- The "hidden" attribute can be set to "true" to indicate
that the -->
                <!-- field should be hidden. All field attributes will be set
after the -->
                <!-- form is received and before anyone views it. -->
                <!-- If the hidden attribute is omitted, it defaults to
"false". -->
                <FormField name="HiddenData1" hidden="true" />

                <!-- Fields can be given an initial value -->
                <!-- using the optional "Value" element. -->
                <FormField name="ApplicantFirstName">
                    <Value>Jill</Value>
                </FormField>

                <!-- All fields listed anywhere within the FormStructure
element will be -->
                <!-- included in the field data optionally returned by
DownloadDocument. -->
                <!-- To include a field without setting any other property, -
->
                <!-- just give its name. -->
                <FormField name="ApplicationID" />

                <!-- Additional FormField elements, if needed, appear here. -
->

                <!-- Indicates a document section. The optional "name"
attribute -->
                <!-- allows the section to be referenced in subsections and
signature lines. -->
                <Section name="InvestmentData">

                    <!-- Each field to be updated is specified in a FormField
element. -->
                    <!-- The required "name" attribute is the field name in
the document. -->
                    <!-- (A Field name should appear at most once per
UpdateFields element.) -->
                    <!-- It is an error to refer to a field in more than one
FormSection. -->
                    <FormField name="InvestmentOptions" />
                    <!-- The field can be given an initial value -->
                    <!-- using the optional "Value" element. -->
                    <FormField name="Term">
                        <Value>3</Value>
                    </FormField>

                    <!-- Fields may have explicit validations. -->
                    <FormField name="ApplicantEmail">

                        <!-- Optional validation. -->
                        <!-- Note that adding a validation effectively makes
the FormField -->
                        <!-- Mandatory/Required, since the user will be
forced to enter a -->
                        <!-- matching value in order to proceed. -->
                        <Validation>
                            <!-- Required regular expression that the new
data must match. -->
                            <!-- Use the JavaScript (ECMA-262) regular
expression format. -->
                            <Match>^[^@]+\@[^@.]+\.[^@]+$</Match>

                            <!-- Required message to display if the
validation fails. -->
                            <ErrorMessage>Please enter a valid email
address.</ErrorMessage>
                        </Validation>

                    </FormField>

                    <!-- Additional FormField elements, if needed, appear
here. -->

                </Section>

                <!-- The form can contain multiple sections. -->
                <Section name="OfficeUse">
                    <!-- ... -->
                </Section>

                <!-- Additional Section elements, if needed, appear here. -->

            </FormStructure>

            <!-- Indicates that a party must sign the document -->
            <SignatureLine>

                <!--refers to a MemberInfo above -->
                <MemberInfoNumber>1</MemberInfoNumber>

                <!-- Name of the PDF digital signature field -->
                <SignField>Signix_Sig_A_Signature1</SignField>

                ...

                <!-- Indicates the document section that will be updated for
this signer. -->
```

```
                <!-- When the workflow passes a signature line, the values of
-->
                <!-- all fields covered by that line are frozen. -->
                <!-- The optional "editInDocument" attribute can be set to
true to indicate -->
                <!-- that the section is editable in the document. -->
                <!-- Optional, defaults to false. If false, -->
                <!-- then no editing will be allowed, but the covered fields
-->
                <!-- will still be frozen when the signature is created. -->
                <Covers section="InvestmentData" editInDocument="true" />

                <!-- The Covers element may be repeated if the signature
covers multiple -->
                <!-- sections, some of which this signer may edit and some of
which -->
                <!-- they cannot. -->
                <Covers section="OfficeUse" />

                <!-- More Covers elements, if needed. -->

            </SignatureLine>

            <!-- Additional SignatureLine elements may appear here. -->

            ...

        </Form>

        <!-- Additional Form elements, if needed, appear here. -->

    </Data>
</SubmitDocumentRq>
```

Examples by Use Case

Editable Field Variations – Multiple Signatures for One Signer

Any variation of the following fields may be found in the <Section name>:

- TextField1 = Came in with value and is Optional
- TextField2 = Came in with value and is Mandatory/Required
- TextField3 = Came in without value and is Optional
- TextField4 = Came in without value and is Mandatory/Required
- TextField5 = Came in with value, needs validation, and is Mandatory/Required
- TextField8 = Came in without value, needs validation, and is Mandatory/Required

```
<FormStructure>
    <Section name="FormSectionOne">
    <FormField name="TextField1">
        <Value>Samples</Value>
    </FormField>
    <FormField name="TextField2">
        <Value>000135624</Value>
        <Required>true</Required>
    </FormField>
    <Section name="FormSectionOne">
        <FormField name="TextField3">
            <Value></Value>
        </FormField>
        <FormField name="TextField4">
            <Value></Value>
            <Required>true</Required>
        </FormField>
        <FormField name="TextField5">
            <Value>02/04/1996</Value>
            <Validation>
                <Match>^(0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])[-
/.](19|20)\d\d$</Match>
                <ErrorMessage>Please enter a valid date in mm/dd/yyyy
format.</ErrorMessage>
            </Validation>
        </FormField>
        <FormField name="TextField6">
            <Value></Value>
            <Validation>
                <Match>^(0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])[-
/.](19|20)\d\d$</Match>
                <ErrorMessage>Please enter a valid date in mm/dd/yyyy
format.</ErrorMessage>
            </Validation>
        </FormField>
    </Section>
</FormStructure>

<SignatureLine>
    <MemberInfoNumber>1</MemberInfoNumber>
    <SignField>Signature1</SignField>
    <DateSignedField>Signature1Date</DateSignedField>
    <DateSignedFormat>MM/dd/yy</DateSignedFormat>
</SignatureLine>
<SignatureLine>
    <MemberInfoNumber>1</MemberInfoNumber>
    <SignField>Signature2</SignField>
    <DateSignedField>Signature2Date</DateSignedField>
    <DateSignedFormat>MM/dd/yy</DateSignedFormat>
    <Covers section="FormSectionOne" editInDocument="true"></Covers>
</SignatureLine>
<SignatureLine>
    <MemberInfoNumber>2</MemberInfoNumber>
    <SignField>Signature3</SignField>
    <DateSignedField>Signature3Date</DateSignedField>
    <DateSignedFormat>MM/dd/yy</DateSignedFormat>
</SignatureLine>
```

Editable Fields With Values & Anonymous Fields – Multiple Signatures for One Signer

In this example there are some fields that need to be edited by a Signer, but there are also fields that need to be initialized with data but should not be editable by the Signer. We would like to get back the values of all these fields when the completed forms are retrieved via DownloadDocument at the end of the process.

If your fields should be split in both editable and non-editable, please proceed with this example. If your fields should not be editable, please look at “Anonymous Fields Only – No Editable Fields”, below.

```
<FormStructure>
        <FormField name="Account Number">
                <Value>55555</Value>
        </FormField>
        <FormField name="Branch Number">
                <Value>12345</Value>
        </FormField>
        <Section name="FormSectionOne">
                <FormField name="TextField1">
                        <Value>Samples</Value>
                </FormField>
                <FormField name="TextField2">
                        <Value>000135624</Value>
                </FormField>
                <FormField name="TextField3">
                        <Value>02/04/1996</Value>
                        <Validation>
                                <Match>^(0[1-9]|1[012])[- /.](0[1-9]|[12][0-
9]|3[01])[-
                                        /.](19|20)\d\d$</Match>
                                <ErrorMessage>Please enter a valid date in
mm/dd/yyyy format.</ErrorMessage>
                        </Validation>
                </FormField>
        </Section>
</FormStructure>

<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature1</SignField>
        <DateSignedField>Signature1Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
</SignatureLine>
<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature2</SignField>
        <DateSignedField>Signature2Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
        <Covers section="FormSectionOne" editInDocument="true"></Covers>
</SignatureLine>
```

Anonymous Fields Only - No Editable Fields

Recommended if there are no fields that need to be edited, but the values of your fields still need to be included in the DownloadDocument response.

If you have no editable fields, please proceed with this example. If some fields should be editable, please look at “Editable Fields With Values & Anonymous Fields – Multiple Signatures for One Signer”, above.

```
<FormStructure>
        <FormField name="Account Number">
                <Value>55555</Value>
        </FormField>
        <FormField name="Branch Number">
                <Value>12345</Value>
        </FormField>
        <Section name="FormSectionOne">
        </Section>
</FormStructure>

<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature1</SignField>
        <DateSignedField>Signature1Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
</SignatureLine>
```

Editable Fields Without Values – Optional/Signer Can Input

Recommended if field data is not sent with the Form but you want to give the Signer the option to fill in data during Signing Ceremony.

If the fields are blank and can get input by one Signer, please proceed with this example. If the fields are not blank, please look at “Editable Fields With Values – Optional/Signer Can Overwrite Input”, below.

```
<FormStructure>
        <Section name="FormSectionOne">
                <FormField name="Nick Name">
                        <Value></Value>
                </FormField>
                <FormField name="Other Amount Requested">
                        <Value></Value>
                </FormField>
                <FormField name="Desired Funding Start Date">
                        <Value></Value>
                </FormField>
        </Section>
</FormStructure>

<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature1</SignField>
        <DateSignedField>Signature2Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
        <Covers section="FormSectionOne" editInDocument="true"></Covers>
</SignatureLine>
```

Editable Fields With Values – Optional/Signer Can Overwrite Input

Recommended only if data being over-written is not integral to processing the form.

If Signer removes the value of the field and does not replace it, is that okay? If yes, proceed with this example. If no, look at “Editable Fields Without Values – Mandatory/Signer Has To Input”, below.

```
<FormStructure>
        <Section name="FormSectionOne">
                <FormField name="First Name">
                        <Value>Linda</Value>
                </FormField>
                <FormField name="Last Name">
                        <Value>Jones</Value>
                </FormField>
                <FormField name="Desired Funding Start Date">
                        <Value>10/26/2020</Value>
                </FormField>
        </Section>
</FormStructure>
```

```
<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature1</SignField>
        <DateSignedField>Signature2Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
        <Covers section="FormSectionOne" editInDocument="true"></Covers>
</SignatureLine>
```

Editable Fields Without Values – Mandatory/Signer Has To Input

Recommended if data does not come with the Form and Signer's input is integral to processing the form.

If the Signer doesn't input data in the field and is then blocked from signing until they do, is that okay? If yes, proceed with this example. If no, look at “Editable Fields Without Values – Optional/Signer Can Input”, above.

```
<FormStructure>
        <Section name="FormSectionOne">
                <FormField name="First Name">
                        <Value></Value>
                        <Required>true</Required>
                </FormField>
                <FormField name="Last Name">
                        <Value></Value>
                        <Required>true</Required>
                </FormField>
                <FormField name="Desired Funding Start Date">
                        <Value></Value>
                        <Required>true</Required>
                </FormField>
        </Section>
</FormStructure>

<SignatureLine>
        <MemberInfoNumber>1</MemberInfoNumber>
        <SignField>Signature1</SignField>
        <DateSignedField>Signature2Date</DateSignedField>
        <DateSignedFormat>MM/dd/yy</DateSignedFormat>
        <Covers section="FormSectionOne" editInDocument="true"></Covers>
</SignatureLine>
```

Supported Field Types

This section describes the supported field types in more detail.

Text Fields

Text fields work as shown in the examples above.

Check Box Fields

A Check Box field represents a single or a group of related/unrelated buttons/boxes which are independent of each other. Each Check Box operates individually, so a Signer can toggle each response on/off. Signer can select more than one option in a grouping.

If you have a grouping of buttons/boxes, for which there can only be one choice checked off please look at Radio Buttons instead.

Example of Check Boxes:

Question = Which colors do you like?

- □ Blue
- □ Yellow
- □ Red
- □ Green

Signer could choose none, 1st, 2nd, 3rd, any combination of those, or all of the checkboxes as their answer.

Check boxes have two states, on and off.

If the <Value> element is included, use "Yes" for the on state and "Off" for the off state. The PDF document must also define the value of the field's on state as "Yes" and the value of the off state as "Off".

A <Validation> element may be included to require the user to check or uncheck the box. <Match>Yes</Match> will require it to be checked, <Match>Off</Match> will require it to be unchecked. Omit the <Validation> element to allow either state.

Here is an example for a checkbox field named "AcceptTerms" that is off by default and that the user is required to check:

```
<FormField name="AcceptTerms">
    <Value>Off</Value>
    <Validation>
```

```
        <Match>Yes</Match>
        <ErrorMessage>You must accept the terms to complete this
transaction.</ErrorMessage>
    </Validation>
</FormField>
```

Radio Button Fields

A Radio Button field represents a group of related and mutually exclusive on/off buttons. Radio button grouping allows for toggling between any of the buttons in the group.

In a PDF, radio buttons can be configured to either allow or disallow “toggle to off”. Conventionally, once a radio button is set all you can do is set a different one. There is no way to unset all of them. However, if the PDF is configured to allow toggle to off, then you can click a set button to unset it, leaving no buttons selected.

In conventional radio buttons, each button represents a different choice and there can only be one button per choice. With PDF radio buttons, however, multiple buttons can represent the same choice. If the “radios in unison” option is on, then when any one of these buttons is selected or cleared, they all are. This facility can be useful if, for example, you want the same choice to be available on multiple pages.

If you have a grouping of buttons or boxes where more than one choice can be selected, please look at Check Boxes instead.

Example of Radio Buttons:

Question = Which is your one favorite color?

- Blue
- Yellow
- Red
- None of the Above

Signer must choose 1st, 2nd, 3rd or 4th as their answer only.

Warning Notes: Some Radio Buttons are set up by PDF forms providers to look like Radio Buttons but are in fact Check Boxes with some nonstandard rules or settings to make them act like Radio Buttons. Dependent on the settings there may be conflicts that may impede the proper use of Radio Buttons in our system, since our system adheres to the radio button specification as described on page 441 of the PDF standard: https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf. Conflict may be limited by making sure that the “Radio” flag and other settings are correct for those fields.

Each button in a PDF Radio Button group has an on value and an off value. The on values are whatever the PDF defines them to be. The off value is normally "off". The value of a Radio

Button field is the value of the button in the on state. If no buttons are on, then the value of the group is "off".

If the <Value> element is included, use "OFF" or 'off' to turn off all the buttons, or a button value to turn on a specific button.

All the Radio Buttons in a grouping must have the same name.

Examples by Use Case

A value is required and an initial choice is already selected

The checkbox or boxes corresponding to the given value will be selected by default. (In this example, “1” might be the value associated with the first button, “2” the value of the second button, etc.)

```
<FormField name="1own.ExemptPayeeCodeType">
        <Value>1</Value>
        <Required>true</Required>
</FormField>
```

A value is required but nothing is selected before the user click a choice themselves

```
<FormField name="1own.ExemptPayeeCodeType">
        <Value>OFF</Value>   <!-- can use ‘0’, ‘off’ or ‘OFF’ -->
        <Required>true</Required>
</FormField>
```

Nothing is selected before the user clicks a choice, and they may elect not to choose

To make Radio Buttons ‘Optional’, just leave out <Required>:

```
<FormField name="1own.ExemptPayeeCodeType">
        <Value>OFF</Value>     <!-- can use ‘0’, ‘off’ or ‘OFF’ -->
</FormField>
```

The user is required to choose, and gets a field-specific error message if they don’t

A <Validation> element may be included to require the user to select a particular button. Conditional Validation is not currently supported.

Here is an example for a radio button field named "AcceptTerms" that is “NotAccepted” by default, and that requires the user to select “Accepted”:

```
<FormField name="AcceptTerms">
    <Value>NotAccepted</Value>
    <Validation>
        <Match>Accepted</Match>
        <ErrorMessage>You must accept the terms to complete this
transaction.</ErrorMessage>
    </Validation>
</FormField>
```

Here is an example if want to insure that the Signer selects Yes or No for a non-defaulted radio button (a radio button that is not yet selected yes or no). The error message will be displayed in red on a dialog box when none of the radio buttons is selected:

```
<FormField name="RadioButtonGroup">
    <Value>OFF</Value>
        <Validation>
                <Match>[yY]es\[nN]o</Match>
                <ErrorMessage>Please select a radio button.</ErrorMessage>
        </Validation>
</FormField>
```

Optional Features

Editable Fields Focus

If you want this feature, request SIGNiX to turn it on for you. Note that each SubmitDocumentReq call will be reviewed as it is received to confirm that it is compatible with this feature (see Important Note below).

What it does: During Signing the Signer is typically taken to every “Optional” and “Mandatory” field until they reach the Signature. In some cases, clients may find that the Signers have already read through the documentation and just want to come back and fill out the ‘Mandatory’ fields and just sign. This feature will take the Signer directly to the first ‘Mandatory’ field and sequentially to the next one until reaches the Signature assigned to those fields. If all of the “Mandatory” fields have been addressed, the Signer is allowed to Sign. If a “Mandatory” field was missed, a popup will warn the Signer that they still have action to take before can be considered complete.

Important Note: If you are using ‘Mandatory’ fields and you want to use the Editable Fields Focus feature, all <FormField name> elements belonging to a <SignatureLine> must sit above that Signature block. If you are unable to break these elements up into sections to sit above each Signature, assign <Section name> to the very last <SignatureLine> or <Initial Line> for the feature to work. If unable to do that, your API will not be able to take advantage of the feature.

If using “Mandatory” Radio Buttons and you want to use the Editable Fields Focus feature, they have to be set up in the PDF as Radio Buttons and not as Check Boxes that look like Radio Buttons. If they are set up as Check Boxes and the following example is true, the Signer may be forced to check off all of the Check Boxes in a grouping before being allowed to sign. Obviously, this results in Signer confusion over the process and the Recipient of data not being able to determine what the Signer really wanted. Example: Display a ‘Male’ or ‘Female’ Radio

Button grouping. The Signer can answer only one of the two buttons, but is forced to check off both because both are set to ‘Mandatory’.

Field Data Report

The signed documents are returned to the client in the response to the DownloadDocument request (see /B2BSpec/ for more detail).

The DownloadDocument request also allows the caller to optionally request that document field data be included in the response. When requested, the response will contain the value of each field listed in the <FormStructure> section of the original SubmitDocument request.

```
<?xml version="1.0" ?>
<DownloadDocumentRq
        xmlns="!urn:com:signix:schema:sdddc-1-1">
        <CustInfo>
                <Sponsor>oem</Sponsor>
                <Client>oemclient</Client>
                <UserId>oemclient</UserId>
                <Pswd>your-password-here</Pswd>
        </CustInfo>
        <Data>
                <DocumentSetID>-147fb68b:1043a5db9ca:-
7f5fid11or</DocumentSetID>

                <!-- Determines whether a field data report will be included
in the response. -->
                <!-- Optional. Defaults to false. -->
                <IncludeFieldData>true</IncludeFieldData>

                ...

        </Data>
</DownloadDocumentRq>
```

The DownloadDocument response will include the field data when requested. The format of the returned data matches the content of the "fields" element of XFDF (see /XFDF/).

```
<?xml version="1.0" ?>
<DownloadDocumentRs
        xmlns="!urn:com:signix:schema:sdddc-1-1">
        <Status>                          <!-- occurs once only -->
                <StatusCode>0</StatusCode>   <!-- one of the numeric SC_*
values -->
                <StatusDesc>success</StatusDesc> <!-- human-readable message -
->
        </Status>
        <Data>                        <!-- occurs when Status.StatusCode is 0 -
->
                <TransactionID>aefd45aa-6a36-4ab5-8cc8-
4dc6e895fa21</TransactionID>
```

```
                <FileName>J-Smith_02242005.zip</FileName> <!-- optional, max 
255 -->

                <!-- The order of the Form elements matches the order in the -
->
                <!-- original SubmitDocument request -->
                <Form>                          <!-- Occurs once for each 
document. -->
                        <Desc>Express Application #1 </Desc>
                        <FileName>J-
Smith_02242005_032557_Express_Application_axx339.pdf
                        </FileName>
                        <MimeType>application/pdf</MimeType>  <!-- the type of 
document -->
                        <Length>5631</Length>     <!-- length of decoded Data 
in bytes -->

                        <!-- Field data report from this document. -->
                        <!-- Present if IncludeFieldData was true in the 
request. -->
                        <Fields>
                                <Field name="ApplicantSSN">
                                        <Value>000456789</Value>
                                </Field>

                                <!-- Other Field elements, if needed -->

                        </Fields>

                        <!-- the bytes of the entire document, encoded in 
Base64 format -->
                        <Data>JVBERi0xLjQNJeLjz9MNCj ... 
YgMCBvYmogPDwvTGluZWFy</Data>

                </Form>

                <!-- Additional Form elements, if needed, appear here. -->
        </Data>
</DownloadDocumentRs>
```

References

/B2BSpec/ "SIGNiX B2B Interface Guide" Version 4.0, July 2018. Available at https://www.signix.com/development-community-documentation

/XFDF/ Adobe. "XML Forms Data Format Specification" Version 2.0. September 2007.