SAML SINGLE SIGN-ON API

AND

SAML ORDER INTEGRATION API

August 2024

Xpressdocs Partners, Ltd. 1301 NE Loop 820, Fort Worth, TX 76137, USA

+1 817.547.9743 | www.xpressdocs.com

Contents

1 Introduction

SAML is an XML-based open standard data format for exchanging authentication and authorization data between parties. SAML enables web browser single sign-on, allowing the seamless sharing of services between companies while utilizing the user's company authentication servers. This eliminates the need for users to remember additional logins or passwords when accessing services, such as Xpressdocs, through their browsers.

This technical document outlines the protocol details necessary to effectively utilize the services offered by Xpressdocs for SAML Single Sign-On System. It is designed to provide guidance to developers and administrators seeking to implement and integrate SAML-based authentication solutions.

1.1 Acronyms and Definitions

This section provides definitions for all acronyms and terms introduced in this document.

Acronym	Definition
SAML	Security Assertion Markup language
SP	Service Provider: Xpressdocs sign-on application
IdP	Identity Provider: vendor's authentication service application
Users	Users are the base-level entities in the system, representing individuals attempting to gain authentication in Xpressdocs. Each user is typically associated with a single office, where they perform their roles and access system resources. However, certain users, such as office admins, may be associated with multiple offices by passing in multiple officeIds (see section 4.4), granting them access to several office locations. In addition to office associations, users may also be associated with multiple regions passing in multiple regionIds. The userId uniquely identifies each user in the system, and they possess attributes such as personal information (e.g., name, email) and role-specific permissions. The system ensures that users have the appropriate level of access based on their associations with offices and regions, facilitating effective role-based access control.
Offices	Offices are entities representing location-based groupings of users within the system. Each office contains specific address information and serves as the primary organizational unit where users are associated. An office is typically linked to one region. The officeId uniquely identifies each office within the system. Only one region may be associated with an office, ensuring that each office falls under a specific geographical or organizational area. Specific users (e.g., office admins) may have access to multiple offices.
Regions	Regions are optional entities that group offices into larger, typically geographical, areas. Each region encompasses one or more offices, providing a way to manage offices on a broader scale. Region admin users may be linked to multiple regions, giving them access to all offices within those regions. The regionId uniquely identifies each region in the system. If an office is not associated with a specific region, the regionId field may be left empty.
Vendor	The provider of single sign on or order integration services

1.2 Assumptions

1. All SAML timestamp values must be compliant to W3C XML schema data type specification and must be expressed in UTC with no time zone component.
2. The ID on all SAML identifiers such as assertions, requests, and responses should be unique to avoid conflicts.
3. The SP server's time and IdP server's time will be in sync to use the response timestamp for validation. The time should be synchronized against the time.nist.gov NTP server.

1.3 Xpressdocs URLs for SP-initiated and response POST

Stage or testing URL:

https://development.awsdev.xpressdocs.com/next/sso/saml.php?company=<providedByXD>*

Production URL:

https://www.xpressdocs.com/next/sso/saml.php?company=<providedByXD>*

1.4 Xpressdocs URLs for IdP-initiated POST

Stage or testing URL:

https://development.awsdev.xpressdocs.com/next/sso/saml_idp.php?company=<providedByXD>*

Production URL:

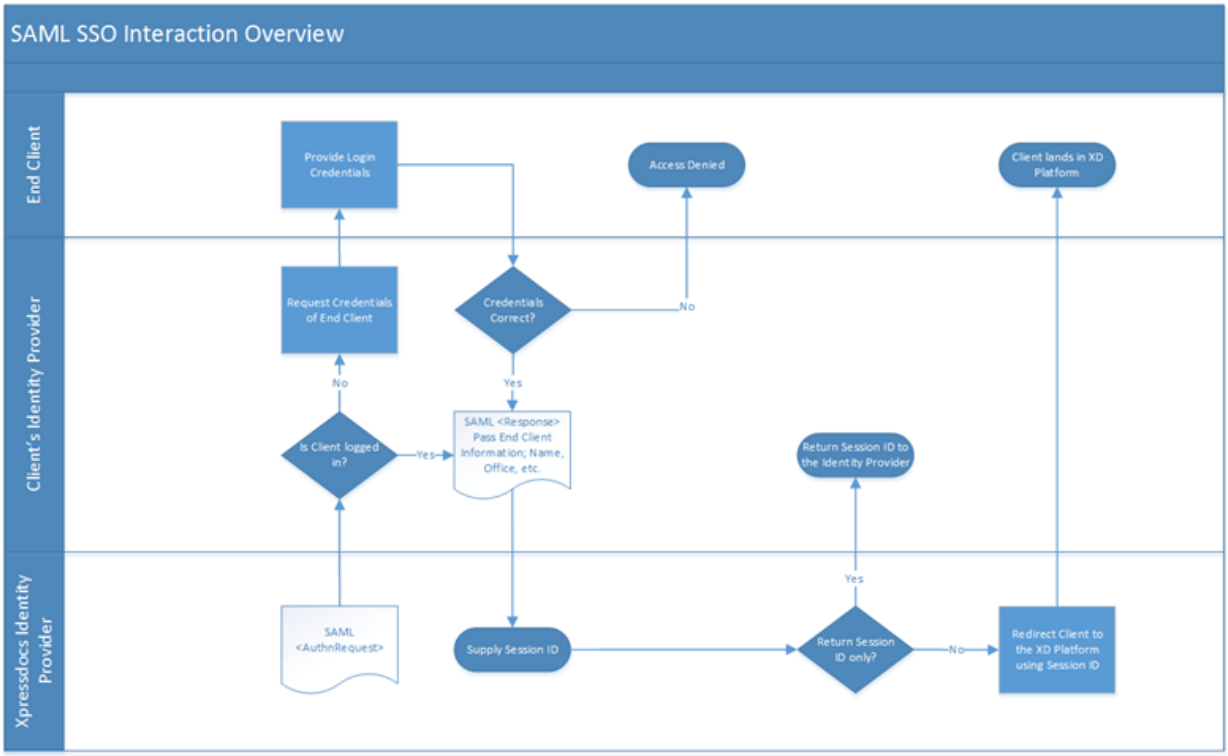
https://www.xpressdocs.com/next/sso/saml_idp.php?company=<providedByXD>*

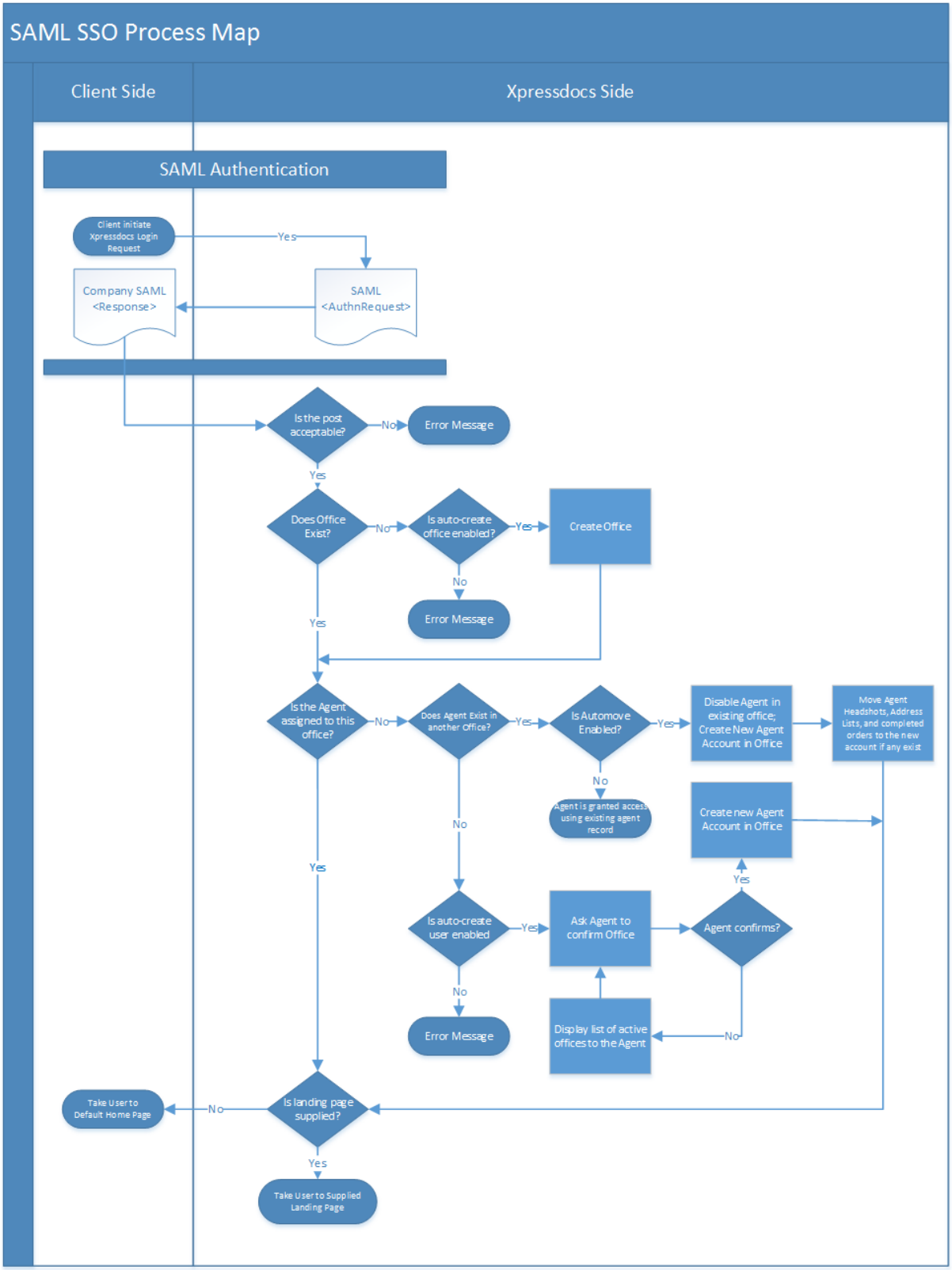
*additional saml_id param may be required depending on existing SSO configurations

2 SAML Integration Flow

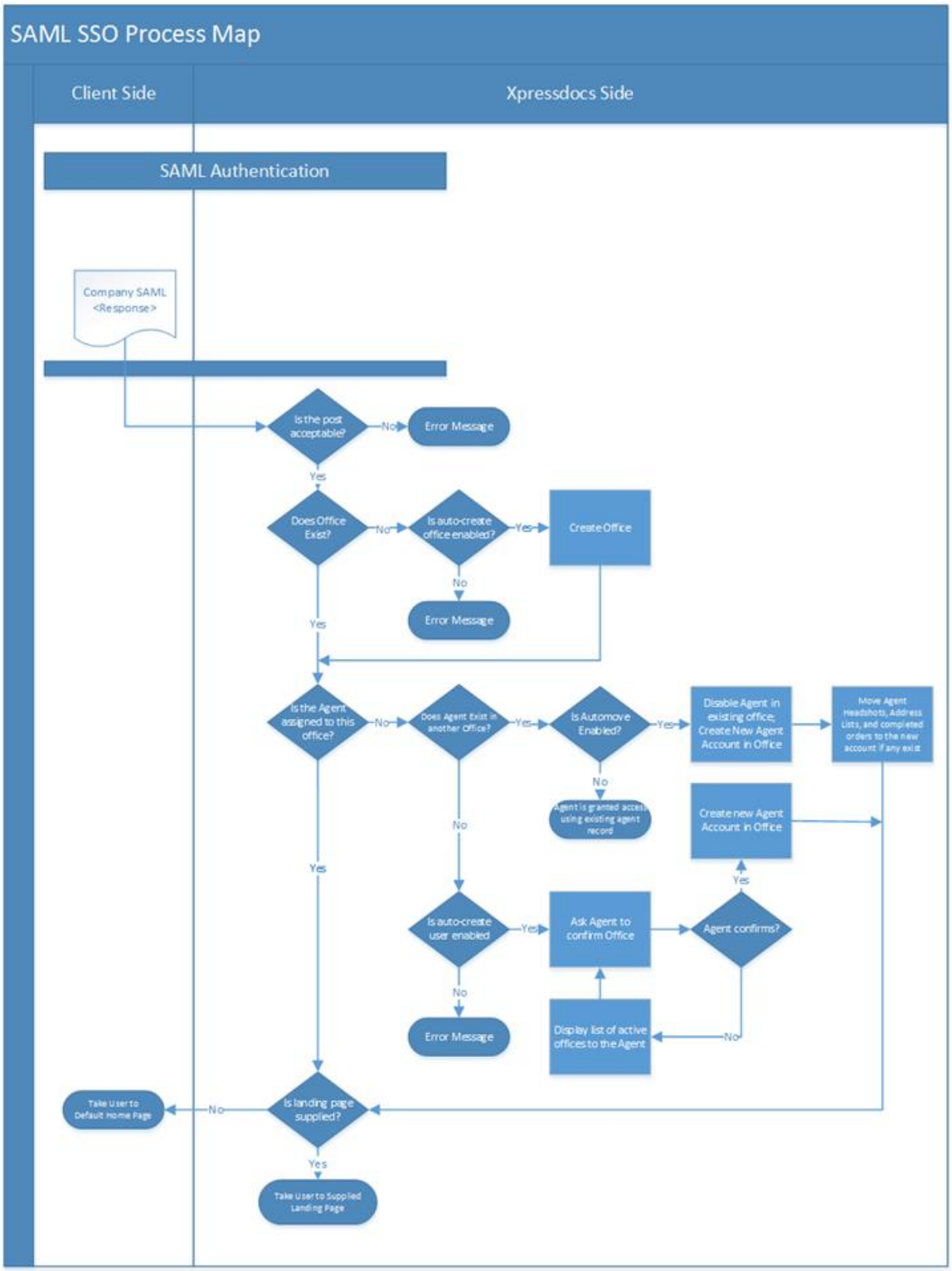
2.1 SAML SSO Process Maps

SP-Initiated SAML Overview




SAML SSO Process Map
Client Side
Xpressdocs Side
SAML Authentication
Client initiate Xpressdocs Login Request
Yes
Company SAML <Response>
SAML <AuthnRequest>
Is the post acceptable?
No
Error Message
Yes
Does Office Exist?
No
Is auto-create office enabled?
Yes
Create Office
No
Error Message
Yes
Is the Agent assigned to this office?
No
Does Agent Exist in another Office?
Yes
Is Automove Enabled?
Yes
Disable Agent in existing office; Create New Agent Account in Office
Move Agent Headshots, Address Lists, and completed orders to the new account if any exist
No
Agent is granted access using existing agent record
Create new Agent Account in Office
Yes
No
Yes
Is auto-create user enabled
Yes
Ask Agent to confirm Office
Agent confirms?
No
Error Message
Display list of active offices to the Agent
No
Take User to Default Home Page
No
Is landing page supplied?
Yes
Take User to Supplied Landing Page

IdP-Initiated SAML and validation flow Overview



2.2 SAML SSO Process Map Steps

1) Process begins with SP-initiated or IdP-initiated SAML and IdP sends a SAML response

SP-initiated (Preferred)	**IdP-initiated**
User initiates a connection to Xpressdocs through a link on their company website	User initiates a connection to IdP through a link on their company website
SP looks up IdP and sends an AuthnRequest (Section 3)	
IdP responds with a SAML response to SP-initiated URL	IdP sends a SAML response to IdP-initiated URL

2) Xpressdocs validates the SAML response and compares it against the data we currently have on the user.
   a) If the post is corrupt or otherwise unacceptable an error message is displayed to the user
3) Xpressdocs checks if the user's office exists.
   a) If the office does not exist, the system will attempt to auto-create the office.
      i) If auto-create office is disabled an error will be displayed, "*Error Code: SSO-206 Attempt to create Office account or Login was not successful. Contact your account manager at Xpressdocs for assistance. 1.866.977.3627*"
      ii) If auto-create office is enabled, Xpressdocs will create the office.
4) Xpressdocs checks to see if the user exists within the user's office.
   a) If the user does not exist within the user's office, Xpressdocs checks to see if the user exists.
      i) If the user does not exist, Xpressdocs will attempt to auto-create the user.
         (1) If auto-create user is disabled an error will be displayed, "*Error Code: SSO-207 Attempt to create User account or Login was not successful. Contact your account manager at Xpressdocs for assistance. 1.866.977.3627*"
         (2) If auto-create user is enabled, Xpressdocs will create the user.
   b) If the user exists, Xpressdocs will attempt to move the user to the user's office.
      i) If auto-move is disabled, then the user will log in to their existing office in the Xpressdocs system.
      ii) If auto-move is enabled, the user is moved to the new office along with their headshots, address lists, and completed orders.
5) Xpressdocs checks to see if a Landing Page URL was supplied with the Login Request
   a) If not, the user is taken to the default home page
   b) If so, the user is taken to the specified landing page

3 AuthnRequest (SP-initiated only)

After the user initiates a connection to Xpressdocs through a link on their company website, Xpressdocs SP will send a AuthnRequest to the IdP Sign-On Service URL. The following section describes the requirements of the call.

3.1 RelayState Param

RelayState is a form post parameter which provides information about the user trying to login. This RelayState will be passed back to Xpressdocs along with SAML response without any modification or inspection.

3.2 SAMLRequest Param

SAMLRequest is a form post parameter that contains SAML 2.0 AuthnRequest XML base64 encoded.

3.3 Authentication Request XML

The initial service request to the Identity provider should be submitted via an HTTP POST method with the Attributes of:

Element Name	Description	Possible Values	Required
ID	This unique Id generated by the Xpressdocs when creating the AuthnRequest. This ID will be referenced in the SAML response's attribute "InResponseTo"	_5348301c-0016-476e-8b2d-117be490b50d	Yes
Destination	The URL of the Sign-On Service at the Identity provider	https://idp.example.org/SAML2/SSO/Artifact	Yes
Issuer	Application Id that will be provided by the Identity Provider.	https://sp.example.com/SAML2	Yes
ProtocolBinding	Protocol to use to transport AuthnRequest and response.	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST	Yes
AssertionConsumerServiceURL	Location to which the SAML Response will be sent to.	https://sp.example.com/SAML2/SSO/Artifact	Yes

```
<saml2p:AuthnRequest
  ID="_5348301c-0016-476e-8b2d-117be490b50d"
  Version="2.0"
  IssueInstant="2012-03-01T16:31:40.403Z"
  Destination="https://idp.example.org/SAML2/SSO/Artifact"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  AssertionConsumerServiceURL="https://sp.example.com/SAML2/SSO/Artifact"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
>
  <saml2:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    6F9514DA-107F-475E-9EEF-C3B07FA7521A
  </saml2:Issuer>
  <saml2p:NameIDPolicy />
</saml2p:AuthnRequest>
```

4 SAML Response

4.1 SAML Response Values

The vendor will send a SAMLResponse to Xpressdocs. SAMLResponse is a form post parameter. The elements/attributes contained in the SAMLResponse are as follows:

Fields marked with an asterisk (*) are required if the auto-update user or office pref is set to YES

Element Name	Description	Required
InResponseTo	SP-initiated – use ID from SAMLRequest IdP-initiated – do not include	For SP
Destination	POST destination URL	Y
StatusCode	Status code (urn:oasis:names:tc:SAML:2.0:status:Success)	Y
NotBefore	SAML response is valid from this time.	Y
NotOnOrAfter	SAML response is valid until this time.	Y
Assertion Attributes : UserID	User Id: Unique Identifier of the user record, used for identifying the user	Y
Assertion Attributes : Username	User Name associated with User Id	N
Assertion Attributes : Email	Email Address of the user	Y
Assertion Attributes : FirstName*	First name of the associated user	Y
Assertion Attributes : MiddleName	Middle name of the associated user	N
Assertion Attributes : LastName*	Last Name of the associated user	Y
Assertion Attributes : DirectPhone	Phone Number of the user	N
Assertion Attributes : DirectPhone2	Secondary Phone Number of the user	N
Assertion Attributes : License	License Number for user	N

Assertion Attributes : Url	Web page URL for user	N
Assertion Attributes : HeadshotUrl	URL for headshot image to upload for user's headshot gallery	N
Assertion Attributes : Role	Permission level for the User - Highest Level: Company, Company Admin Middle Level: Branch, Region, Office, Division Base Level: Agent (Default if role excluded)	N
Assertion Attributes : AgentDisplay1	Override Default Display Values (Default: 'FirstName' 'LastName')	N
Assertion Attributes : AgentDisplay2	Override Default Display Values	N
Assertion Attributes : AgentDisplay3	Override Default Display Values	N
Assertion Attributes : AgentDisplay4	Override Default Display Values (Default: 'DirectPhone')	N
Assertion Attributes : AgentDisplay5	Override Default Display Values (Default: 'DirectPhone2' or 'OfficePhone')	N
Assertion Attributes : AgentDisplay6	Override Default Display Values (Default: 'License')	N
Assertion Attributes : AgentDisplay7	Override Default Display Values (Default: 'Email')	N
Assertion Attributes : AgentDisplay8	Override Default Display Values (Default : 'Url')	N
Assertion Attributes : OfficeId	Unique Identifier of the office record, used for identifying the office in Identity Provider System.	Y
Assertion Attributes : OfficeIds	Comma-separated office IDs for Office Admin	N
Assertion Attributes : OfficeName*	Office name	Y
Assertion Attributes : OfficeLegalName	Office legal name if different than office name	N
Assertion Attributes : OfficeAddress1*	Address1 field	For New
Assertion Attributes : OfficeAddress2	Address2 field	N
Assertion Attributes : OfficeCity*	City	For New
Assertion Attributes : OfficeState*	State	For New
Assertion Attributes : OfficeZip*	Zip	For New
Assertion Attributes : OfficeCountry	2 Digit Country Code (Default: 'US')	N
Assertion Attributes : OfficePhone*	Office Phone number	For New
Assertion Attributes : OfficeEmail	Office Email Address	N
Assertion Attributes : OfficeFax	Office fax number	N
Assertion Attributes : OfficeDisplay1	Override Default Display Values (Default: 'OfficeLegalName' or 'OfficeName')	N
Assertion Attributes : OfficeDisplay2	Override Default Display Values (Default: 'OfficeAddress1' 'OfficeAddress2')	N
Assertion Attributes : OfficeDisplay3	Override Default Display Values (Default: 'OfficeCity', 'OfficeState' 'OfficeZip')	N
Assertion Attributes : OfficeDisplay4	Override Default Display Values (Default: 'OfficePhone')	N
Assertion Attributes : OfficeDisplay5	Override Default Display Values (Default: 'OfficeFax')	N
Assertion Attributes : OfficeDisplay6	Override Default Display Values	N

Assertion Attributes : OfficeDisplayCustom1	Override Custom Display Value	N
Assertion Attributes : OfficeDisplayCustom2	Override Custom Display Value	N
Assertion Attributes : RegionId	Unique Identifier for the region records	N
Assertion Attributes : RegionIds	Comma-separated region IDs for Region Admin	N
Assertion Attributes : RegionName	Name identifying the region	N
Assertion Attributes : LandingPageURL	Page to drop user onto following a successful login: • (Default) Home - /app/ • APM Properties - apm2_properties.php • APM Property Marketing Program Preferences - apm_profile.php • My Account - /app/account/ • My Listings - /app/listings • Order History - /app/account/orders/history • Specific Category - /app/cat/<idCategory> • Specific Subcategory - /app/cat/<idCategory>/sub/<idSubCat> • Specific Template - /app/product-options/<idTemplate>	N

4.2 Sample SAML Response XML

Note: The content in the sample SAML response is generated based on the Schema defined in SAML OASIS Documentation.

Please refer to the above table for the important attribute and elements that contains the assertion values in the SAML response.

For best quality, associated headshot URLs should reference the image directly and not an image manipulation utility or other image proxy. If a proxy is required, the quality should be verified early in the integration process; high-resolution images are preferred. Please note that for printed images to be sharp without any blur or pixelation an image with at least 300 dpi is required.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:Response
 ID="_03013930-4bd3-4ce9-8462-b3865792bffd"
 InResponseTo="_5348301c-0016-476e-8b2d-117be490b50d"
 Destination="http://www.example.com/ProcessSamlResponsePage.aspx"
 xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
 IssueInstant="2012-03-02T16:09:16.425Z"
 Version="2.0"
```

```
>
 <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  http://www.example.com/IDProvider/
 </saml2:Issuer>
 <saml2p:Status xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
 </saml2p:Status>
 <saml2:Assertion
  ID="_a999fd99-44f6-42a2-8033-46393aa58789"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
  IssueInstant="2012-03-02T16:09:16.425Z"
  Version="2.0"
 >
  <saml2:Issuer>http://www.example.com/Idprovider/</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
   <ds:SignedInfo>
     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
     <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
     <ds:Reference URI="#id1452300820598262945394l9">
       <ds:Transforms>
         <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
         <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
           <ec:InclusiveNamespaces PrefixList="xs"
             xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         </ds:Transform>
       </ds:Transforms>
       <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
       <ds:DigestValue>VpOL06gxis+Lk7dMNmDiYDZhEEdv2rAePTBXHit6Ia0=</ds:DigestValue>
     </ds:Reference>
   </ds:SignedInfo>
    <ds:SignatureValue>Sign Here</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>Certificate Here</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <saml2:Subject>
   <saml2:NameID>Jane.Doe@domain.com</saml2:NameID>
   <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml2:SubjectConfirmationData
     Recipient=" http://www.example.com/ProcessSamlResponsePage.aspx"
    />
   </saml2:SubjectConfirmation>
```

```
  </saml2:Subject>
  <saml2:Conditions NotBefore="2012-03-02T15:59:16.425Z" NotOnOrAfter="2012-03-
02T16:19:16.425Z" />
  <saml2:AuthnStatement AuthnInstant="2012-03-02T16:09:16.425Z">
   <saml2:AuthnContext>
    <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml2:AuthnCon
textClassRef>
   </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
   <saml2:Attribute Name="UserID">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">12345</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="Email">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">Jane.Doe@domain.com</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="FirstName" >
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">Jane </saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="LastName">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">Doe</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="OfficeId">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">12345ABCD</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="Role">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">Agent</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute Name="Landing_Page_URL">
    <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xsd:string">/app/account/orders/history</saml2:AttributeValue>
   </saml2:Attribute>
  </saml2:AttributeStatement>
 </saml2:Assertion>
</saml2p:Response>
```

4.3 Security

To enhance the security and integrity of the SAML Response exchanged between the vendor and Xpressdocs, it is recommended to sign the SAML Response using a valid X.509 certificate. This digital signature ensures the authenticity and non-repudiation of the SAML Response data.

Steps to Sign the SAML Response:

1. **Obtain a Valid X.509 Certificate:** The vendor must acquire a valid X.509 certificate from a trusted Certificate Authority (CA) or create a self-signed certificate to sign the SAML Response. The certificate should be securely stored and managed to maintain its confidentiality and integrity.
2. **Share your public certificate with Xpressdocs:** The vendor will provide their public X.509 certificate for storage in the Xpressdocs system.
3. **Sign the SAML Response:** Before sending the SAML Response to Xpressdocs, apply the digital signature using the private key associated with the X.509 certificate. The signature should be included in the SAML Response metadata to verify the authenticity of the data.
4. **Verify the Signature:** Xpressdocs will validate the digital signature using the public key of the X.509 certificate provided by the vendor. If the signature verification succeeds, Xpressdocs can trust the integrity of the SAML Response and proceed with further processing. Xpressdocs will also check that the certificate matches the stored certificate.

By signing the SAML Response with a valid X.509 certificate, vendors can bolster the security of the authentication process and establish a trusted communication channel with Xpressdocs.

4.4 Sample POST example for SAMLResponse

Sample PHP self-posting form. Note params are SAMLResponse and RelayState.

```
$destination = XPRESSDOCS_SAML_URL
$response_encoded = base64_encode( $response );
$form = '
    <form method="post" action="'.$destination.'">
        <input type="hidden" name="SAMLResponse" value="'.$response_encoded.'"/>
        <input type="hidden" name="RelayState" value="'. param('RelayState') .'"
/>
        <input type="submit" name="sub" value="Submit" id="sub"
style="display:none;"/>
    </form>';

echo $form;
echo '<script language="javascript">
    document.getElementById("sub").click();
</script>';
```

The purpose of the PHP code snippet is to generate a self-posting form for submitting the SAMLResponse and RelayState parameters securely.

4.5 Multi Office Support

If a user is associated with multiple offices, the vendor may pass multiple comma-separated values into OfficeIds OR may pass in multiple SAML attribute values within the SAML attribute for OfficeId. Offices must be configured on the Xpressdocs side for this to feature to work properly.

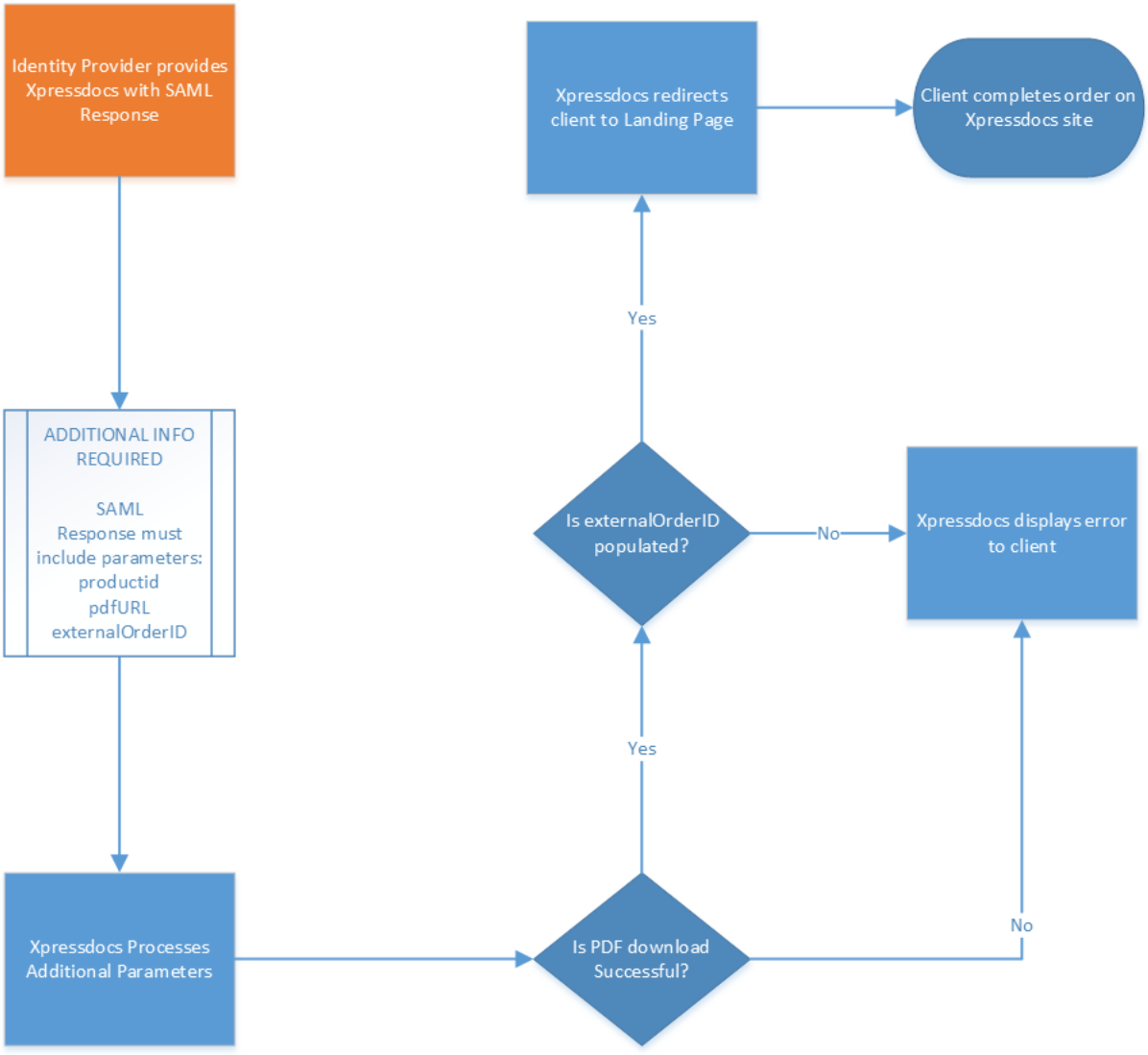
```
<saml2:Attribute Name="OfficeId">
   <saml2:AttributeValue>OfficeId1</saml2:AttributeValue>
   <saml2:AttributeValue>OfficeId2</saml2:AttributeValue>
   <saml2:AttributeValue>OfficeId3</saml2:AttributeValue>
</saml2:Attribute>
```

5 Order Integration (Formerly Order Integration API - OIA)

5.1 Definition

Xpressdocs allows a Vendor to provide a print-ready pdf from a third party design tool into the Xpressdocs Order workflow while using the IdP for authentication. The pdf URL and ancillary data must be provided in the SAML Response as described below. The user will be redirected into the Xpressdocs system. The user will then select additional options and checkout.

5.2 Detailed Order Integration Overview



5.3 Overview

The vendor will call Xpressdocs Order Integration by adding Order Integration parameters to the initial SAML response sent to the link provided by Xpressdocs.

5.4 Request

The vendor will input the order as additional attributes in the initial SAML response:

Field	Description	Required	Value Type
PdfUrl	Vendor location of print-ready-file	YES	String
ExternalOrderId	Vendor's unique order reference number	YES	Alphanumeric or Numeric
ProductId	Product code provided by Xpressdocs	YES	Alphanumeric
TemplateKey	Template key provided by Xpressdocs	No	Alphanumeric
QRRedirectUrl	Location for QR code to redirect to	No	String
QRRedirectType	'url', 'xpresslinks', or 'homevalue'	No	String

```
<saml2:Attribute Name="PdfUrl">
 <saml2:AttributeValue>https://sampleurl.pdf</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="ExternalOrderId">
 <saml2:AttributeValue>123UnqiueNumber</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="ProductId">
 <saml2:AttributeValue>SMPC</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="TemplateKey">
 <saml2:AttributeValue>12345</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="QRRedirectUrl">
 <saml2:AttributeValue>https://www.redirectUrl.com</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="QRRedirectType">
 <saml2:AttributeValue>url</saml2:AttributeValue>
</saml2:Attribute>
```

5.5 Validation of Request

1) Xpressdocs will download the PDF at the provided **PdfUrl**.
   a) Xpressdocs will validate that the pdf URL is accessible. In the event of an error, the user will receive an error message.
2) Next, Xpressdocs will verify the **ExternalOrderId** is not empty.
   a) The vendor must provide a unique reference identifier for each order to ensure tracking of all orders. If the vendor does not include ExternalOrderId, the user will receive an error message.
3) Finally, Xpressdocs will verify that response includes the **ProductId** and/or the **TemplateKey.**
   a) The vendor must provide either the templatekey that associates their template to the corresponding Xpressdocs template or the Xpressdocs productid that is associated with their template.
      i) This allows Xpressdocs to match the pdf to the appropriate product and to manage special case needs such as page orientation.
      ii) If the vendor does not include either a productid or a templatekey, the user will receive an error message.

5.6 Order Creation

Xpressdocs will create a new order and populate the known fields, including the **External Order Id** as provided by the vendor. Xpressdocs will redirect users into the Xpressdocs Platform. The user will then complete the order through the Xpressdocs platform order flow.

5.7 Diagnostics

Xpressdocs logs all responses received from vendors. Xpressdocs recommends that vendors do this as well. Troubleshooting issues is much easier when a clear record of what data was exchanged exists, including error messages and what happened as a result.

5.8 Tasks

Xpressdocs needs to do the following to integrate with a vendor:

1. Xpressdocs must provide vendor with product IDs &/or template keys.
2. Xpressdocs must support both SAML SSO and Order Integration setup with vendor.
3. SAML SSO integration requires validation of SAML attribute exchange.
4. Upon successful testing of SAML integration, then Order Integration testing may begin.
5. Xpressdocs must coordinate both SAML SSO and Order Integration testing with vendor prior to deployment.

6 Links to oasis documents

http://saml.xml.org/saml-specifications