Windsor Solutions



Electronic Signature Agreement Design Guide

Updated for Version 4.18

Revision Date: September 28, 2023



This page intentionally left blank

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Notes |
| 4.9 | 7/16/2020 | Windsor Solutions | Initial version |
| 4.10 | 9/4/2020 | Windsor Solutions | Updated to reflect v4.10 release. |
| 4.11 | 1/29/2021 | Windsor Solutions | Updated to reflect v4.11 release. |
| 4.11 | 2/17/2021 | Windsor Solutions | Updated to reflect new online identity proofing/electronic signature agreement. |
| 4.12 | 7/20/2021 | Windsor Solutions | Updated to reflect v4.12 release. |
| 4.13 | 5/2/2022 | Windsor Solutions | Updated to reflect v4.13 release. |
| 4.14 | 8/31/2022 | Windsor Solutions | Updated to reflect release 4.14. |
| 4.15 | 12/19/2022 | Windsor Solutions | Updated to reflect v4.15. |
| 4.16 | 5/11/2023 | Windsor Solutions | Updated to reflect v4.16. |
| 4.17 | 6/28/2023 | Windsor Solutions | Updated to reflect v4.18. |

**Contents**

[Introduction 1](#_Toc64459442)

[Overview 2](#_Toc64459443)

[Electronic Signature Agreement Form Design 3](#_Toc64459444)

[ESA Formatting/Layout 3](#_Toc64459445)

[Referencing Submission Data 3](#_Toc64459446)

[Testing 5](#_Toc64459447)

[Publishing 6](#_Toc64459448)

# Introduction

The nFORM tool supports the ability to define documents that can be generated from the system. This includes Electronic Signature Agreement (ESA) forms, which are presented to external user during the identity proofing process. These documents are used to authorize online signatures, in preparation for submission and electronic signing.

The content in the ESA forms can be static or dynamically driven based on data available for a user. This capability allows agencies to provide a highly formatted and targeted ESA forms to online users.

This document provides some technical guidance on how to design the ESA form template(s).

Please note that this is an advanced topic and is only to be utilized by advanced technical staff which knowledge of software coding practices.

# Overview

In order to provide clients with ESA forms which uniquely meet their needs, Windsor has implemented a feature that allows system users to:

1. Establish ESA Document Templates for use as the ESA forms for all external users of the system.
2. Access user data for display within the document generated from the ESA Document Templates.

Document Templates are formatted/designed in MS Word, using LINQ coding syntax to access Submission data, Aspose.Words for .NET to compile and generate the document, Aspose.PDF to PDF the document.

ESA forms are based on Document Templates and are made available to users during the registration and identity proofing processes and allow the user to prove their authenticity in order to authorize online signatures.

Document generation supports a highly configurable, dynamic, feature set and as such, requires advanced skills and detailed testing/verification. Customers who elect to define advanced configurations in document templates should consult with Windsor to ensure appropriate conventions and techniques are utilized to ensure a successful implementation.

Two ESA forms are available, depending on the systems configuration, and whether online identity proofing is enabled. These include:

* **Online ESA**: The online ESA is presented to external users during the online identity proofing process and the user must agree to the terms presented in this document via online signature.
* **Mail-In ESA**: The mail-in ESA is presented to external users during the mail-in identity proofing process. The user receives the document via email and must download the document, agree to the terms presented, sign the document and return it to the agency.

Configuration/design of each of these ESA forms is presented in this document.

# Electronic Signature Agreement Form Design

## ESA Formatting/Layout

ESAs are formatted and designed using MS Word (version 2007 or higher). The MS Word document can be highly formatted and include advanced features such as headers, footers, paging, tables, etc.

Be sure to test out the layout and functionality of the template using the publishing and testing instructions in this document.

## Referencing Submission Data

The ESA Document Templates have the ability to access data from the submission for reference on the form. In order to access this data, the system uses Aspose.Words and an Aspose specific variation of the LINQ coding syntax. Here are some helpful resources on the template syntax:

* [https://docs.aspose.com/display/wordsnet/Template+Syntax](https://docs.aspose.com/display/wordsnet/Template%2BSyntax)
* [https://docs.aspose.com/display/wordsnet/Typical+Templates](https://docs.aspose.com/display/wordsnet/Typical%2BTemplates)

### \*\*Disclaimer\*\*

This is an advanced topic and is only to be utilized by advanced technical staff which knowledge of software coding practices.

Staff should know that if errors are encountered while running these documents, it is often due to issues with coding and data being presented not fitting into the assumed structure presented in this document. Most commonly, data referenced may not be present or may be returned in a different format than expected (e.g., text value returned in what was believed to be a numeric field). These situations must be handled/pre-empted to prevent errors during document generation. Professional services may be advisable/required from Windsor to assist in the creation of these documents.

### Data Attributes

The form can access data user and some system attributes (e.g., User Name, First/Last Name, Agency Name, etc.) directly.

#### User Name Example

The following example returns the user’s name.

|  |
| --- |
| <<[Principal.FirstName]>> <<[Principal.LastName]>> |

#### Address Example

In this example, the Template is attempting to access text from the address that the user entered on their user profile. The code below performs the following tasks:

* Defines the **userAddress** variable to target the first address assigned to the user (data model supports multiple addresses, but UI does not).
* Defines the **userAddressStr** variable to simplify the many references to this control.
* The **siteAddressStr** variable is established by concatenating all the constituent parts of the address control (e.g., street address, secondary address, city, state, zip, etc.) and to improve formatting, HTML break tags are added where needed.
* Performs error handling to handle the situation where the user address is null.

See example below:

|  |
| --- |
| <<var [userAddress=Principal.Addresses.FirstOrDefault()]>><<var [userAddressStr = userAddress == null ? “”: userAddress.Street +(userAddress.Street !=null?”<p />”:””) + userAddress.AddressLine2 + (userAddress.AddressLine2 !=null?”<p />”:””) + userAddress.Locality +”, ” + userAddress.AreaCode + “ “ + userAddress.PostalCode] >> |

Note: each individual address component could be referenced separately (versus one address block, if desired.

One defined, the userAddressStr can be referenced from anywhere in the form using the following syntax:

|  |
| --- |
| Address: <<[userAddressStr]:”format” –html >> |

#### Telephone Example

The following returns all telephone numbers assigned to the user.

|  |
| --- |
| <<foreach [item in Principal.Phones]>><<[item.PhoneTypeDescription]>>: <<[item.Number]>><</foreach>> |

#### Email Example

The following returns the Email address assigned to the user.

|  |
| --- |
| <<[Principal.PrimaryEmail]>> |

#### Other Considerations

Below are some additional considerations when presenting dynamic data attributes:

* When defining the variables for the data to be accessed at the top of the template:
	+ Perform this work at the top of the page (prior to the data attribute being referenced)
	+ Do NOT add carriage returns in or between these definitions as the template will include these carriage returns in the presented document.
* If a value is empty (e.g., not available), the value will be returned as a blank to the document. It’s worth noting that in some cases, blank values are saved in controls as a “?” to help with URL encoding. In these cases, the “?” will need to be replaced with a blank value during display.
* Case does matter when referencing variables so be sure that your casing is in sync.
* Configuration values (in the Vars.config, VarsOverrides.config, etc.) can be exposed to the document generator. These can be exposed by referencing the existing configuration properties in the following format:

|  |
| --- |
| <add key="docgen.config.settings" value=" app.title, app.name, message.email.appBaseUrl, esa.recipient.organization.name, esa.recipient.mailing.address, hcs.recipient.organization.name, hcs.recipient.mailing.address"/> |

Any attributes referenced here can be accessed from within the document by referencing the configuration value without periods and capitalizing secondary words. So the app.Title configuration setting will be referenced as:

|  |
| --- |
| <<[appTitle]>> |

The esa.recipient.mailing.address configuration setting will be referenced as:

|  |
| --- |
| <<[esaRecipientMailingAddress]>> |

## Testing

It is recommended that the ESA Document Templates be tested in a test environment, prior to publishing.

### Mail-In ESA Form

Once a draft Mail-In ESA is assigned to the system, a user can download the Mail-In ESA document by selecting the “Download Mail-In Form” option in the Enable Electronic Signatures portion of the registration and identity proofing process.

You can also force the download of the Mail-In ESA Form by assembling a URL to generate and download the ESA document directly. To do this assemble the URL in the following format:

[*root application URL*]/ElectronicSignatureAgreement/[*UserID*]

For example, in Windsor’s QA environment, for a specific user, the URL would look like the following:

[**https://qaweb.windsorsolutions.biz/nForm/ElectronicSignatureAgreement/71fcc80b-5d8d-4571-84f4-28c60bc19944**](https://qaweb.windsorsolutions.biz/nForm/ElectronicSignatureAgreement/71fcc80b-5d8d-4571-84f4-28c60bc19944)

If issues are encountered and the Mail-In ESA Form returns a runtime error, further details about the nature of the error can be found in the application logs which can be found here, [*root web application folder*]\App\_Data\logs where *root web application folder* is equal to the location of the web application.

### Online ESA Form

Once a draft Online ESA is assigned to the system, a user can download the Online ESA document by selecting the “Use Digital Authentication Service (recommended)” option in the Enable Electronic Signatures portion of the registration and identity proofing process. The document will be presented within the page as a part of the identity proofing and in addition, the document will be emailed to the user once identity proofing is completed.

You can also force the download of the Online ESA Form by assembling a URL to generate and download the ESA document directly. To do this assemble the URL in the following format:

[*root application URL*]/DocumentGeneration/ [*UserID*]?template=IdentityProofingCromerrAgreement

For example, in Windsor’s QA environment, for a specific user, the URL would look like the following:

[**https://qaweb.windsorsolutions.biz/nForm/DocumentGeneration/71fcc80b-5d8d-4571-84f4-28c60bc19944?template=IdentityProofingCromerrAgreement**](https://qaweb.windsorsolutions.biz/nForm/DocumentGeneration/71fcc80b-5d8d-4571-84f4-28c60bc19944?template=IdentityProofingCromerrAgreement)

If issues are encountered and the Online ESA Form returns a runtime error, further details about the nature of the error can be found in the application logs which can be found here, [*root web application folder*]\App\_Data\logs where *root web application folder* is equal to the location of the web application.

## Publishing

### Mail-In ESA Form

To publish an established Mail-In ESA template, perform the following:

* Save the template with the name ElectronicSignatureAgreementTemplate.docx
* Copy/Move the template to your agencies Override folder, [*root web application folder*]\DocumentTemplates\Word where *root web application folder* is equal to the location of the web application.
Note: any changes made to the Mail-In ESA template should be backed up and provided to Windsor to ensure the master template is included in the client build, to ensure it is not overwritten with an old version during the next deployment process.
* At this point, the template will be in use.

Please note that the base Template that is shared throughout the system must be named **ElectronicSignatureAgreementTemplate.docx**. Also note that if the template is opened by a user, other users will not be able to download the template.

### Online ESA Form

To publish an established Online ESA template, perform the following:

* Save the template with the name IdentityProofingCromerrAgreement.docx
* Copy/Move the template to your agencies Override folder, [*root web application folder*]\DocumentTemplates\Word where *root web application folder* is equal to the location of the web application.
Note: any changes made to the Online ESA template should be backed up and provided to Windsor to ensure the master template is included in the client build, to ensure it is not overwritten with an old version during the next deployment process.
* At this point, the template will be in use.

Please note that the base Template that is shared throughout the system must be named **IdentityProofingCromerrAgreement.docx**. Also note that if the template is opened by a user, other users will not be able to download the template.