Perceptive eAuthorize

Installation and Setup Guide

Version: 3.1.x

Written by: Product Knowledge, R&D
Date: October 2024

Hyland

Documentation Notice

Table of Contents

About Perceptive eAuthorize

The Perceptive eAuthorize solution enables you to send documents for electronic signature to any authorized signatory, including non-Perceptive Content users. You can send documents, residing within or outside of Perceptive Content, to a single or multiple authorized signatories whose signatures are required. The signed documents upload automatically and are stored in the Perceptive Content repository. If the signatory declines the document, you receive an email notification. A signatory can also forward the email with the link to the documents to any person inside or outside your organization.

This solution offers the following advantages.

- The electronic process of signing saves you time.
- Anyone inside or outside your organization can sign documents.
- Signatories do not need access to Perceptive Content.
- Signatories receive email notifications of documents to sign. The email contains a link to the document and instructions. Alternatively, documents can be immediately presented on the user's screen. You can send a single document or an envelope containing multiple related documents.

The following three levels of eAuthorize are available.

- **Standalone**. AssureSign is sold as a standalone solution, with no integration with Pereptive Content.
- **Post-signature integration**. A non-Perceptive Content document is provided to AssureSign for signature, is signed, and then the signed document is uploaded into Perceptive Content.
- **Full integration**. A Perceptive Content document is submitted to AssureSign for signature, is signed, and then the signed document is uploaded to Perceptive Content.

This document provides the configuration and setup guidelines for Perceptive Content Connector for AssureSign.

Important After you finish installing and configuring eAuthorize, complete the steps in the *Perceptive eAuthorize Getting Started Guide* to test your configuration and confirm that the installation and setup is successful.

For information about using Perceptive eAuthorize, see the *Perceptive eAuthorize Getting Started Guide*.

Requirements

Software prerequisites

Before you run the eAuthorize solution, ensure that your system meets the following prerequisites.

- Perceptive Connect Runtime 2.2.x is installed.
- Content Connector 2.2.x is installed and configured to use the Perceptive Integration Server instance where the eAuthorize components are to be installed.

Notes

- Ensure that all the Content Connector related components are in an active state.
- Perceptive Content Server and Client 7.3 or higher are installed and running properly.
- You can log into a Perceptive Content user account with manager privileges.

- You have administrator access to the AssureSign environment to create and edit AssureSign templates.
  - Sandbox environment: https://sb.assuresign.net/documents/Default.aspx
  - Production environment: https://na1.assuresign.net/Login.aspx or the URL for the local instance of AssureSign
- Ensure that the appropriate IP ranges for AssureSign are open for inbound and outbound communiation. The currently used IP addresses are available in https://support.assuresign.net/hc/en-us/articles/224274907.
- Ensure that the appropriate ports used by PCR are open for communication with AssureSign.

License

The following licenses are required to run eAuthorize.

- Perceptive eAuthorize
- Integration Framework version 7.3 or higher
- Perceptive Content Server, version 7.3 or higher
- Perceptive Content Client, version 7.3 or higher
- Integration Server for Apps version 7.3 or higher, and Transaction Pack
- iScript
- Optional. eForms and Doc Control Suite

Install the eAuthorize components

Download the eAuthorize files

To obtain Perceptive product installation files, contact the Hyland Software Technical Support group. For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.

Extract the eAuthorize files

To extract the eAuthorize files, open the eAuthorize zip file and extract the contents to a temporary directory, such as <**EAUTHORIZEINSTALLDIR**>**.**

Install the Perceptive Content files

To install the Perceptive Content files, complete the following steps.

1. Copy all the files from **<EAUTHORIZEINSTALLDIR>/inserver/script** to the **<IMAGENOWDIR6>/script**. This folder contains the following iScript files.
   - CreateTask.js creates a task in Perceptive Content if a document fails in workflow.
   - DocumentStoreAndForward.js stores and forwards a document in workflow queue.
   - RefreshAuditHistory.js refreshes Audit History.

- eAuthorize folder containing IN_WorksheetManager.jsh, IN_XML.jsh, INBasePath.jsh, and Util_Misc.jsh.

2. Copy all the files from **<EAUTHORIZEINSTALLDIR>/ inserver/etc**. to the **<IMAGENOWDIR6>/etc** folder. This folder contains the **eAuthorize** folder, which contains the xml file **eAuthorize_config.xml**.
3. Copy the **<EAUTHORIZEINSTALLDIR>/eAuthorize** folder to the **<IMAGENOWDIR6>** folder**.** This folder contains the following files:
   - **ConfigurePerceptiveContentForEAuthorize.js** creates custom properties, task templates, and eForms in Perceptive Content.
   - **CreateWorkflowForEAuthorize.js** creates a default workflow for the eAuthorize process.
   - **logger.jsh** is an include file required for the above .js files**.**
   - forms with the following subfolders and files:
     - agent
       - CallAssureSign.js
     - AuditHistory
       - data_definition
     - AssureSign_AuditHistory.xml
     - Presentation
       - AssureSign_AuditHistory.xsl
       - Asynch.js
       - Forms.css
       - Refresh_16.png
       - Section_header_bg.gif
     - ImmediatePresentment
       - data_definition
       - AssureSign_ImmediatePresentment.xml
       - Presentation
         - AssureSign_ImmediatePresentment.xsl
         - Forms.css
         - Forms.js
         - Section_header_bg.gif

Install eAuthorize connector

Before you install the connector, ensure the following prerequisites are met.

- Perceptive Connect Runtime is installed and running.
- Content Connector is installed and running in Perceptive Connect Runtime.

- Perceptive Content Server is installed and running.
- Integration Server is installed and running.

To Install eAuthorize connector, complete the following steps.

1. In a browser, go to the **Perceptive Connect Runtime Dashboard** at **http://{Connect Runtime host name}:{port}**.
2. In the browser dialog box, in the **User name** and **Password** fields, enter the user name and password for Connect Runtime.

   Note The default user name and password are **admin**. However, an administrator can change the defaults during the Connect Runtime installation process.
3. In the left pane, under **Manage**, click **Install a Connector**.
4. On the **Upload New Bundles** page, from **<EAUTHORIZEINSTALLDIR>**, drag the **eAuthorize-install-<version>.zip** file over to the right side of the page and drop it**.**
5. When the installation completes, click **Accept** to accept the installation or **Roll back** to undo the installation. You must accept or roll back the installation before PCR can process the next item.

Note For more information on installing Connectors, refer to the Install connectors section of the *Perceptive Connect Runtime Installation Guide*.

Setup the environments

Setup Perceptive Content environment

Create the default components

To create the default Perceptive Content forms, custom properties, and task templates, run the following command from the **<IMAGENOWLOCALDIR6>/bin64** folder**.**

```
$ intool.exe --cmd run-iscript –-file
../eAuthorize/ConfigurePerceptiveContentForEAuthorize.js
```

Note See Appendix A: Manually Create Perceptive Content Components for instructions on manually setting up the task templates and eforms.

Create the default workflow

To create the default workflow, run the following command from the **<IMAGENOWLOCALDIR6>/bin64** folder**.**

```
$ intool.exe --cmd run-iscript –-file ../eAuthorize/CreateWorkflowForEAuthorize.js
```

Notes

- When running this javascript file, the system creates a file, **<IMAGENOWLOCALDIR6>/eAuthorize/EAuthorizeConnectQIds.txt**, which contains the IDs of the three created ConnectQs. These IDs are used when creating the Perceptive Connect Runtime channels for eAuthorize.
- See Appendix A: Manually Create Perceptive Content Components for instructions on manually setting up the workflow.

Create document types

Document types must match AssureSign Templates. The requirements for the document type are different for documents originating from Perceptive Content and documents originating from a third party application.

You must configure documents originating from a third-party applications, including forms-server, as described below.

The name of the document type **must** contain the words "**AssureSign**" and "**External**". This is the default document type for document transmission

The following custom properties are required:

- AS_AUTH_TOKEN
- AS_ID

Documents originating from Perceptive Content must have a document type name that matches an AssureSign template name. Add the appropriate custom properties from the list.

Required properties:

- AS_AUTH_TOKEN
- AS_FAILURE_NOTIFICATION_TYPE
- AS_ID
- AS_KEEP_ORIGINAL_DOCUMENT
- AS_SIGNATURE_STATUS

Optional properties based on AssureSign template requirements:

- Signatory 1 Email Address
- Signatory 1 Full Name
- Signatory 1 First Name
- Signatory 1 Last Name

Note To add more custom properties, see the About custom properties for multiple signatures section.

Create folders

You can send multiple documents for signature at the same time by sending a folder of documents to be signed through an AssureSign envelope. The folder must have a folder type name that matches an AssureSign envelope template name. To create a folder type to send multiple documents for signing, create a Folder Type with the following properties:

Required properties:

- AS_AUTH_TOKEN
- AS_FAILURE_NOTIFICATION_TYPE
- AS_ID
- AS_KEEP_ORIGINAL_DOCUMENT
- AS_SIGNATURE_STATUS

Setup Perceptive Connect Runtime (PCR) environment

Configure AssureSign

To configure AssureSign, complete the following steps.

1. In the **Perceptive Connect Runtime Dashboard**, under **Manage**, click **Configure**.
2. Under **Configure eAuthorize**, click **Configure AssureSign**.
3. In the **Configure AssureSign** dialog box, complete the following sub-steps:
    1. In the **WSDL URL** field, enter the appropriate AssureSign WSDL URL.
        - For Sandbox, enter the location https://sb.assuresign.net/Documents/Services/DocumentNOW/v2/DocumentNOW.svc?wsdl.
        - For production instance, enter the location https://na1.assuresign.net/Documents/Services/DocumentNOW/v2/DocumentNOW.svc?wsdl
        - If you use the locally installed AssureSign application, enter the following location [Site Root]/AssurSign/services/documentnow/v2/documentnow.svc?wsdl

          Note For AssureSign local, the machine name and port number are where the local instance of AssureSign is installed.
    2. In the **Local domain name** field, enter the name where AssureSign is installed locally. You need to populate this field for on premise AssureSign installation. Leave this field blank if you are using an AssureSign cloud environment (sandbox or production).
    3. In the **User name** field, enter the AssureSign account user name with which you submit documents for signature.
    4. In the **Context ID** field, enter the **AssureSign Account Context ID** provided on the **AssureSign Settings** page for the **DocumentNOW Account Context Identifier**.
    5. In the **Template tag** field, enter the **AssureSign** template tag. You can provide any value (string). Use this value as the template tag when you create a template in AssureSign. For details, see the AssureSign Quick Reference Guide for instructions to create a template and the template tag.

    Note If the template tag value the in the Configuration page of Perceptive Connect Runtime dashboard does not match the same in the AssureSign template, documents cannot be submitted for signature.
4. Click **Save**.

Configure email

To configure email, complete the following steps.

1. In the **Perceptive Connect Runtime Dashboard**, under **Manage**, click **Configure**.
2. Under **Configure eAuthorize**, click **Configure Email**.
3. In the **Configure Email** dialog box, complete the following substeps:
    1. In the **SMTP server name** field, enter the SMTP server name.
    2. In the **SMTP server port** field, enter the SMTP server port.
    3. In the **Sender email ID** field, enter the SMTP server sender's email ID.

4. Select **SMTP Authentication** if authentication is required.
5. In the **Authentication user name** field, enter the SMTP server authentication user name.
6. In the **Authentication password** field, enter the SMTP server authentication password.
7. Select **Start TLS** if the SMTP server requires TLS encryption.
8. In the **Encryption Protocols**, if required, enter a space delimited list of supported protocols to negotiate when connecting to the email server.
9. Select **Check Server Identity** to enforce certificate validation of your SMTP server.

4. Click **Save**.

Verify the connector JAR bundle status

To verify that the JAR bundle is correctly uploaded to Perceptive Connect Runtime, complete the following steps.

1. In the **Perceptive Connect Runtime** dashboard, under **Troubleshoot**, click **List OSGi Bundles**.
2. On the **OSGI Bundles** page, under the **Name** column, look for the following bundle names and ensure that the status for all the bundles is **Active**. Active status indicates the bundle is started successfully.
   - eAuthorize Configurations
   - eAuthorizeAssureSignConnector
   - eAuthorizeCommon
   - eAuthorizeImageNowConnector
   - eAuthorizeSignatureEndpoint
3. Under **Troubleshoot,** click **List OSGi Components**.
4. On the **OSGI Components** page, under the **Name** column, look for the following components and ensure that the status of all the components is **Active**. Active status indicates that the component is hosted successfully.
   - eAuthorize AssureSign Connector
   - eAuthorize Download Signed Document Action
   - eAuthorize eForm Processor
   - eAuthorize ImageNow Connector
   - eAuthorize REST Endpoint
   - eAuthorize Send To AssureSign Action
   - eAuthorize SignatureEndpoint
   - eAuthorize Submit For Signature Action
   - eAuthorizeASConfiguration
   - eAuthorizeEmailConfiguration

Configure the Content Connect service

To configure the Content Connect service, complete the following steps.

1. Navigate to the location the **<IMAGENOWDIR6>\etc** directory where Content Server is installed.
2. Open the **inserverWorkflow.ini** file in a text editor and in the [General] section, configure the following settings.
   - Set **connect.uri** to your Connect Runtime instance in the following format.

     ```
     http://{Connect Runtime host name}:{port}/rs/workflowTrigger
     ```

   - Set **connect.timeout** to your desired expiration time.
3. Save the file and close the text editor.
4. Optional. The system automatically loads the setting after a short period of time. If you want to reload the configuration immediately, restart the Perceptive Content Workflow Agent service.

Create and configure channels

To create and configure channels using Perceptive Connect Runtime, complete the following steps.

1. In the **Perceptive Connect Runtime** dashboard, under **Manage**, select **Create a channel**.
2. In the **Name** field, enter a channel name for **SubmitForSignature**.
3. In the **Trigger** list, select **Integration ASQ Trigger**.

   Notes

   - If Content Connector is not installed and configured, **Integration ASQ Trigger** will not appear in the **Trigger** list.
   - If you installed the default workflow, the IDs are located in the**<IMAGENOWLOCALDIR6>/eAuthorize/EAuthorizeConnectQIds.txt** file. You can also find the IDs in properties of the queue available in the **Perceptive Content Workflow Designer**.
4. In the **Workflow Queue ID** field, type the Connect queue ID of the **SubmitForSignature** queue that you previously created.
5. Click **Continue**.
6. **Select action …** under **Actions**. Then select **SubmitForSignatureAction** under e**AuthorizeSignatureEndpoint**.
7. Click **Save Inputs**.
8. Click **Enable Channel** and then click **OK** in the confirmation dialog box.
9. Under **Manage**, click **Create a channel** and enter a channel name for **SendToAssureSign**.
10. In the **Trigger** list, select **Integration ASQ Trigger**.
11. In the **Workflow Queue ID** field, type the Integration queue ID of the **SendToAssureSign** that was created previously.
12. Click **Continue**.
13. Under **Actions**, click **Select action**.
14. Under e**AuthorizeSignatureEndpoint**, select **SendToAssureSignAction**.

15. Click **Save Inputs**.
16. Click **Enable Channel** and then click **OK** in the confirmation dialog box.
17. Under **Manage**, click **Create a channel** and then enter a channel name for **DownloadSignedDocuments**.
18. In the **Trigger** list, select **Integration ASQ Trigger**.
19. In the **Workflow Queue ID** field, type the Integration queue ID of the **DownloadSignedDocumentChannel** queue that was created previously.
20. Click **Continue**.
21. Under **Actions**, click **Select action**.
22. Under **eAuthorizeSignatureEndpoint**, select **DownloadSignedDocumentsAction**.
23. Click **Save Inputs**.
24. Click **Enable Channel** and then click **OK** in the confirmation dialog box.

Setup AssureSign environment

Ensure that the following components are set up in Perceptive Content

The eAuthorize ConfigurePerceptiveContentForEAuthorize script sets up several aspects of Perceptive Content to get you started. These automatic configurations are detailed in Appendix B: About the eAuthorize Perceptive Content configuration. However, before you can send a document to AssureSign for signing, you must configure the following Perceptive Content components.

- Task Templates
- Workflow
- Document Types
- Folder Types

Note Task Templates and Workflow are created by ConfigurePerceptiveContentForEAuthorize.js and CreateWorkflowForEAuthorize.js iScripts.

Set up AssureSign DocumentTRAK for status notifications

Log into AssureSign to configure the following DocumentTRAK web notifications. If you are using a local instance of the AssureSign environment, see the Appendix D: Copy web notification templates for information on copying AssureSign web notification templates from the sandbox environment.

eAuthorizeDocumentStatus

To configure eAuthorizedDocumentStatus, complete the following steps.

1. On the **Administration** tab, on the left, click **DocumentTRAK** and then click **Notification Administration**.
2. Under **WEBHOOKS**, copy the **eAuthorize Document Status (Template)** webhook template and then under **General Information**, edit the following information.
    1. In the **Webhook Name** field, enter a name for the webhook.

2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.

3. Click **Continue** three times.

4. Verify that the content of the XML file is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusDocument>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<DOC_STATUS>[Document Status]</DOC_STATUS>
</esig:UpdateStatusDocument>
</soapenv:Body>
</soapenv:Envelope>
```

5. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example. If you make any changes to the **Value** field, click **Save Validator**.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusDocumentResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusDocumentResponse> </soap:Body> </soap:Envelope>
```

6. Click **Finish**.

eAuthorizeStepStart

To configure eAuthorizedStepStart, complete the following steps.

1. On the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration**.

2. Under **WEBHOOKS**, copy the **eAuthorize Step Start (Template)** webhook template and then under **General Information**, edit the following information.

    1. In the **Webhook Name** field, enter a name for the webhook.

    2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to this: **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.

3. Click **Continue** three times.

4. Verify that the XML is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusStepStarted>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<DOC_SIGNING_STEP>[Signing Step]</DOC_SIGNING_STEP>
</esig:UpdateStatusStepStarted>
</soapenv:Body>
```

```
</soapenv:Envelope>
```

5. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example. If you make any changes to the **Value** field, click the **Save Validator** button.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusStepStartedResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusStepStartedResponse> </soap:Body> </soap:Envelope>
```

6. Click **Finish**.

eAuthorizeStepComplete

To configure eAuthorizedStepComplete, complete the following steps.

1. On the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration**.
2. Under **WEBHOOKS**, copy the **eAuthorize Step Complete (Template)** webhook template and then under **General Information**, edit the following information.
    1. In the **Webhook Name** field, enter a name for the webhook.
    2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to this: **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.
3. Click **Continue** three times.
4. Verify that the XML is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusStepCompleted>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<DOC_SIGNING_STEP>[Signing Step]</DOC_SIGNING_STEP>
</esig:UpdateStatusStepCompleted>
</soapenv:Body>
</soapenv:Envelope>
```

7. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example. If you make any changes to the **Value** field, click the **Save Validator** button.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusStepCompletedResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusStepCompletedResponse> </soap:Body> </soap:Envelope>
```

8. Click **Finish**.

eAuthorizeStepUserVisit

To configure eAuthorizedStepUserVisit, complete the following steps

1. On the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration**.
2. Under **WEBHOOKS**, copy the **eAuthorize Step User Visit (Template)** webhook template and then under **General Information**, edit the following information.

1. In the **Webhook Name** field, enter a name for the webhook.
2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to this: **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.

3. Click **Continue** three times.
4. Verify that the XML is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusStepLandingPageVisited>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<DOC_SIGNING_STEP>[Signing Step]</DOC_SIGNING_STEP>
</esig:UpdateStatusStepLandingPageVisited>
</soapenv:Body>
</soapenv:Envelope>
```

5. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example. If you make any changes to the **Value** field, click the **Save Validator** button.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusStepLandingPageVisitedResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusStepLandingPageVisitedResponse> </soap:Body> </soap:Envelope>
```

6. Click **Finish**.

eAuthorizeDocumentCompleted

To configure eAuthorizedDocumentCompleted, complete the following steps.

1. On the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration**.
2. Under **WEBHOOKS**, copy the **eAuthorize Document Completed (Template)** webhook template and then under **General Information**, edit the following information.
    1. In the **Webhook Name** field, enter a name for the webhook.
    2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to this: **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.
3. Click **Continue** three times.
4. Replace the original text in the XML with the following highlighted text. Note that this is for indexing a signed document within Perceptive Content. If the document originated within Perceptive Content and you want to maintain the original document index values, leave the field values blank. To do this, select the existing text in the code in place of the highlighted text and click the exact parameter name enclosed in brackets in the **RequestXMLEditing** field.

    Note The values for **<field1>** through **<field5>** can be static text (no brackets) or can contain the name of an AssureSign jotblock within brackets ([JotblockName]) so the jotblock value can be populated as the index field. For any empty field in AssureSign, the corresponding field in Perceptive Content remains unchanged.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusDocumentCompleted>
<DOC_AUTH_TOKEN>[Document AuthToken]</DOC_AUTH_TOKEN>
<DOC_ID>[Document ID]</DOC_ID>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<field1>(Field 1 value)</field1>
<field2>(Field 2 value)</field2>
<field3>(Field 3 value)</field3>
<field4>(Field 4 value)</field4>
<field5>(Field 5 value)</field5>
</esig:UpdateStatusDocumentCompleted>
</soapenv:Body>
</soapenv:Envelope>
```

5. Click **Continue** twice.
6. Click the **Edit validator** ellipsis button, copy the following code into the **Value** field and then click the **Save Validator** button.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusDocumentCompletedResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusDocumentCompletedResponse> </soap:Body> </soap:Envelope>
```

7. Click **Finish**.

eAuthorizeDocumentStatusKBA

To configure eAuthorizedDocumentStatusKBA, complete the following steps.

1. On the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration**.
2. Under **WEBHOOKS**, copy the **eAuthorize Document Status KBA (Template)** webhook template and then under **General Information**, edit the following information.
    1. In the **Webhook Name** field, enter a name for the webhook.
    2. In the **Endpoint (URL)** field, replace **[TODO:Server IP]** with the PCR host IP address or DNS name. The URL should look similar to this: **http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint**.
3. Click **Continue** three times.
4. Verify that the XML is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusDocumentKBA>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<DOC_SIGNING_STEP>[Signing Step]</DOC_SIGNING_STEP>
<DOC_STATUS>[Document Status]</DOC_STATUS>
<SIGNER_AUTH_FAIL_DET>[Signatory Authentication Failure
Details]</SIGNER_AUTH_FAIL_DET>
<SIGNER_KBA_DET>[Signatory KBA Result Details]</SIGNER_KBA_DET>
```

```
<SIGNER_KBA_RES>[Signatory KBA Result]</SIGNER_KBA_RES>
</esig:UpdateStatusDocumentKBA>
</soapenv:Body>
</soapenv:Envelope>
```

5. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ns2:UpdateStatusDocumentKBAResponse
xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature"> <Return>true</Return>
</ns2:UpdateStatusDocumentKBAResponse> </soap:Body> </soap:Envelope>
```

6. Click **Finish**.

Set up an AssureSign template for signature

For details, see the AssureSign Quick Reference Guide for instructions to create a template. However, consider the following points when creating a template.

- Make sure the template name matches the name of the corresponding document type name in Perceptive Content.
- When using a Template Tag, make sure the Template Tag in AssureSign is same as the value that you provided in the **Configure AssureSign** section of the **Configuration** page of **Perceptive Connect Runtime** dashboard.

Add web notifications

In the **Workflow Template**, for **Web Notifications**, add the notifications as shown in the following table.

Note If there are multiple steps (signatories) defined in the template, then each step (Step 1, Step 2, and so on) should have the following assigned design names.

Stage	Timing	Design Name
Document Started	Before Document Started	eAuthorize Document Status
Step 1	Before Step	eAuthorize Step Start
	After Step	eAuthorize Step Complete
	Landing Page Visited	eAuthorize Step User Visit
Document Completed	After Document Completed	eAuthorize Document Completed
Expiration Warning	No notification selected	
Document Expired	After Document Expiration	eAuthorize Document Status
Document Cancelled	Document Cancelled	eAuthorize Document Status
Document Declined	Document Declined	eAuthorize Document Status
Feedback Submitted	No notification selected	

Stage	Timing	Design Name
Authentication Failed	Authentication Failed	eAuthorize Document StatusKBA
KBA Started	KBA Started	eAuthorize Document StatusKBA
KBA Completed	KBA Completed	eAuthorize Document StatusKBA

Add dynamic JotBlock in PDF documents

To add dynamic JotBlocks in a PDF document, you have to change the following setting in the AssureSign environment.

1. In **AssureSign**, on the **Administration** tab, on the left pane, click **Settings**.
2. In **Document Preferences**, point to **Flatten PDF Documents Prior to Processing** and click **Edit**.

   Note If you cannot find this setting in **Document Preferences**, contact your administrator to set this preference to **No**.
3. Select **No** and then click **Save**.

Setup AssureSign with the ability to download signed documents from a third-party

Signed documents submitted by a third-party automatically upload into Perceptive Content after the signatory signs them. These documents do not originate in Perceptive Content. A third-party sends them for signature manually or from another application. The system then uploads the documents directly into Perceptive Content after signing.

Set up AssureSign for document transmission

The steps given below are for the transmission of third-party documents in Perceptive Content. Complete the following steps to set up the AssureSign environment for document transmission.

1. In **AssureSign**, on the **Administration** tab, open **Notifications**, click **DocumentTRAK** and then click **Notification Administration**.
2. On the **Document Transmission** tab, copy the **eAuthorize External Doc Upload (Template)** document transmission template and then under **General Information**, edit the following information.
   1. In the **Webhook Name** field, enter a name for the webhook.
   2. In the **Endpoint (URL)** field, replace [TODO: Server IP] with the PCR hostname or IP address. The URL should look similar to this: http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint.
3. Click **Continue** three times.
4. Replace the original text in the XML with the following highlighted text by performing the following substeps.
   1. Select the existing text in the code in place of the highlighted text and click the exact parameter name enclosed in brackets in the **Request XML Editing** field.
   2. In the **<field1>**, **<field2>**, **<field3>**, **<field4>**, and **<field5>** tags, enter the appropriate index values for the signed document when it comes into Perceptive Content.

Note Value for **<field1>** through **<field5>** can be static text (no brackets) or could contain the name of an AssureSign jotblock within brackets ([JotBlockName]). If any field value is empty, the corresponding field in Perceptive Content appears empty.

3. In the **<documentType>** tag, enter the document type value that you want the signed document to contain.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UploadSignedDocument>
<DOC_ID>[Document ID]</DOC_ID>
<DOC_AUTH_TOKEN>[Document AuthToken]</DOC_AUTH_TOKEN>
<DOC_ORDER_ID>[Order ID]</DOC_ORDER_ID>
<drawer>default</drawer>
<name>[Document Name]</name>
<field1>[Document Name] Field 1</field1>
<field2>[Document Name] Field 2</field2>
<field3>[Document Name] Field 3</field3>
<field4>[Document Name] Field 4</field4>
<field5>[Document Name] Field 5</field5>
<documentType>AssureSign_External_Abhirup</documentType>
<DOCUMENT>[Completed Document]</DOCUMENT>
<EXTERNAL_DOCUMENT_QUEUE>SignatureComplete</EXTERNAL_DOCUMENT_QUEUE>
</esig:UploadSignedDocument>
</soapenv:Body>
</soapenv:Envelope>
```

Notes

- Ensure that the document keys contain unique values for signed documents. This restricts Perceptive Content from appending a signed third-party document.
- Provide the queue name for the **EXTERNAL_DOCUMENT_QUEUE** parameter to download the signed document in that queue in Perceptive Content. The previous example shows the **SignatureComplete** queue.
- If you do not provide a document name for the name parameter, Perceptive Content uses the document ID as the default document name.
- If you do not provide a type for the documentType parameter, the default document type for document transmission is used. For details, see the Create document types to download signed documents submitted by third party applications section of this document.
- When you use an eForm from Perceptive Forms Server, ensure that document type is not specified as "is a form" for document storage in Perceptive Content.
- If you provide a document type, ensure the document type contains **AS_AUTH_TOKEN**and **AS_ID**as the custom properties.

5. Click **Continue** twice.
6. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example.

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
```

```
<ns2:UploadSignedDocumentResponse
            xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature">
<Return>true</Return>
</ns2:UploadSignedDocumentResponse>
</soap:Body>
</soap:Envelope>
```

7. Click **Finish**.
8. Create a template for document transmission. For details,see the AssureSign Quick Reference Guide for instructions to create a template.

   Note Consider the following points when creating the template.

   - In **Workflow Template**, for **Web Notifications**, ensure that no notifications are selected for any of the stages.
   - For **Document Transmission**, select the design name added in this section.

Setup the AssureSign environment for envelopes

This section outlines the steps to set up the AssureSign environment for sending multiple documents in a folder for signature. To see the steps to send multiple documents see the *Perceptive eAuthorize Getting Started Guide*.

Configure DocumentTRAK for the AssureSign envelope template

To configure DocumentTRAK for the AssureSign envelop template, complete the following steps.

1. In **AssureSign**, on the **Administration** tab, open **Notifications**, click **DocumentTRAK** and then click **Notification Administration**.
2. On the **Webhooks** tab, copy the **eAuthorize Envelope All (Template)** web notification template and then under **General Information**, edit the following information.
   1. In the **Webhook Name** field, enter a name for the webhook.
   2. In the **Endpoint (URL)** field, replace [TODO: Server IP] with the PCR hostname or IP address. The URL should look similar to this: http://<Connect Runtime host name>:<port>/ws/SignatureEndpoint.
3. Click **Continue** three times.
4. Verify that the XML is similar to the following example and then click **Continue** twice.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:esig="http://www.perceptivesoftware.com/asq/esignature">
<soapenv:Header/>
<soapenv:Body>
<esig:UpdateStatusEnvelope>
<ENVELOPE_AUTH_TOKEN>[Envelope AuthToken]</ENVELOPE_AUTH_TOKEN>
<ENVELOPE_ID>[Envelope ID]</ENVELOPE_ID>
</esig:UpdateStatusEnvelope>
</soapenv:Body>
</soapenv:Envelope>
```

5. Click the **Edit validator** ellipsis button and then verify that the XML in the **Value** field is similar to the following example.

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<ns2:UpdateStatusEnvelopeResponse
           xmlns:ns2="http://www.perceptivesoftware.com/asq/esignature">
<Return>true</Return>
</ns2:UpdateStatusEnvelopeResponse>
</soap:Body>
</soap:Envelope>
```

6. Click **Finish**.

Create an envelope template

Email notifications are assigned within the Envelope Template and disabled within the individual Document Templates that are included as part of the Envelope. Otherwise, instead of receiving one notification for all documents in the envelope, the signatories receive multiple email notifications for the same document.

1. In the **Envelope Templates** section, click **New**.
2. In the **Name** field, enter a name for the envelope template. Ensure that the template name matches the name of the corresponding folder type in Perceptive Content.
3. In the **Tag** field in AssureSign, enter a value that matches the value that you provided in the Configure AssureSign section. For additional information on the Configuration, see the Appendix C: Configuration.
4. Select an email design set from the **Email Design Set** list and an **Account** from the **Accessibility** list.
5. Click **Save**.

Edit notifications

To edit notifications, complete the following steps.

1. Click **Edit** in front of the new envelope template.
2. Click **Edit Notifications** and add the notifications as shown in the following table.

Stage	Timing	Design Name
Envelope Started	Envelope Started	eAuthorize Envelope All
Envelope Completed	Envelope Completed	eAuthorize Envelope All
Envelope Expired	Envelope Expired	eAuthorize Envelope All
Envelope Cancelled	Envelope Cancelled	eAuthorize Envelope All
Envelope Declined	Envelope Declined	eAuthorize Envelope All

Set up an eForm for Signature from server download location

If you are using an eForm, you do not have to capture the document in Perceptive Content. The signatory receives the form with a Submit button. After the signatory completes the form and clicks the Submit

button, a Begin signing link displays directly within the eForm. When the signatory clicks the Begin signing link within the eForm, the document appears for signing. The eForm data dynamically populates the data on the document. After signing, you can configure the document within the AssureSign template and then automatically upload it into Perceptive Content.

Setup eForm

To setup eForm, complete the following steps.

1. Log into Perceptive Content with managerial privileges, click **Manage** and then complete the following substeps.
    1. In **Management Console**, in the left pane, click **Forms**.
    2. On the **Forms** tab, click **Open Form Designer** and create a form.

       Note For more information on creating forms using Form Designer, see the *Perceptive Content Form Designer Help Topic*.
    3. On the **Forms** tab, click **Manage Form Components**.
    4. On **Presentations**, select the newly created presentation and click **Modify** and complete the following substeps.
        1. In the left pane, click **Files**.
        2. In the right pane, click **Add**, browse to <**IMAGENOWLOCAL6>/eAuthorize/forms/agent** and select **CallAssureSign.js**, and click **OK**.
2. Open the XSL file from the newly created form folder in the **[*drive*:]:\inserver6\form** directory and complete the following substeps.
    1. Add **<script language="JavaScript" src="CallAssureSign.js"></script>** within the head tag to call AssureSign, as shown in the following example.

```
<?xml version="1.0" encoding="windows-1252"?>
<xsl:stylesheet version="1.0"
xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/form">
<html>
<head xmlns="http://www.w3.org/1999/xhtml">
      <script language="JavaScript" src="CallAssureSign.js"></script>
<meta name="generator"
       content="HTML Tidy for Windows (vers 14 February 2006), see
www.w3.org">
</meta>
<title>ImageNow Forms</title>
<meta name="GENERATOR" content="MSHTML 9.00.8112.16455">
</meta>
<meta name="GENERATOR" content="MSHTML 9.00.8112.16455">
</meta>
<meta name="GENERATOR" content="MSHTML 9.00.8112.16455">
</meta>
<meta content="text/html; charset=us-ascii"
       http-equiv="Content-Type"></meta>
<meta name="GENERATOR" content="MSHTML 9.00.8112.16455">
</meta>
</head>
```

2. Add **<input type="button" value="Submit" onclick="getImmediatePresentmentLink()" />** and ** ** within a body tag to place the **Submit** button and the **Begin signing** link in the form body, as shown in the following example.

```
<input type="button" value="Submit" onclick="getImmediatePresentmentLink()" />
<p id="SigningLink"></p>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

3. In the installed PCR directory, navigate to **conf** directory and open **config.properties** file in a text editor.
4. Make the following entries in the **config.properties** file under the section **Cross Origin Resource Sharing**.

```
##############################
# Cross Origin Resource Sharing
##############################
connect.rs.cors.enabled=true
connect.rs.cors.allow.all.origins=true
connect.rs.cors.allow.credentials=false
connect.rs.cors.allowed.origins=*
```

5. You must modify the CallAssureSign.js file. The following URL in the CallAssureSign.js must point to the eForm service running as an eAuthorize component on PCR.
   - xmlHTTP.open("POST", "http://<Connect Runtime host>:<port>/rs/SignatureEndpoint/SubmitToAssureSign", true)

Configure CallAssureSign.js to work with manually designed eForms

Manually designed eForms are eForms that are created using utilities other than the eForm designer utility of Perceptive Content. If you design the eForm manually, you need to make some changes in the CallAssureSign.js script so that it can recognize the inputs and associated value to construct the URL encoded request parameters.The following example displays the Java script template that you must modify to work with a manually created eForm.

```
function getImmediatePresentmentLink() {
    var allElements = document.all;
    for (var i = 0; i < allElements.length; i++) {
        allElements[i].setAttribute("disabled", true);
    }
    var winLoc = window.location.href;
    var tempStr = winLoc.substring(winLoc.lastIndexOf("/"), winLoc.lastIndexOf("."));
    var formName = tempStr.substring(tempStr.indexOf("_") + 1);

    document.getElementById("SigningLink").innerHTML = 'Please wait....The document
isbeing submitted....Your signing link will appear here shortly';
    var strRequest = "ImageNowFormName=" + formName;
    var row = document.getElementsByTagName("tr");
    for (i = 0; i < row.length; i++) {
        var currentItem = row.item(i);
        if (currentItem.childNodes.length < 2) {
            continue;
        }
        var field = currentItem.childNodes.item(0).innerText;
        var dataItem = currentItem.childNodes.item(1).childNodes.item(0);
        var value;
```

```
        if (dataItem.nodeName.toUpperCase() == "INPUT") {
            value = dataItem.value;
        }
        if (dataItem.nodeName.toUpperCase() == "SELECT") {
            var selectedIndex = dataItem.selectedIndex;
            value = dataItem.getElementsByTagName("option")[selectedIndex].innerText;
        }
        strRequest = strRequest + "&" + field + "=" + value;
    }

    var xmlHTTP;
    if (window.XMLHttpRequest) {
        xmlHTTP = new window.XMLHttpRequest;
    }
    else {
        try {
            xmlHTTP = new ActiveXObject("MSXML2.XMLHTTP.3.0");
        }
        catch (ex) {

        }
    }
    xmlHTTP.open("POST", "http://<Connect Runtime
host>:<port>/rs/SignatureEndpoint/SubmitToAssureSign", true);
    xmlHTTP.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded;charset=utf-8");
    xmlHTTP.onreadystatechange = function () {
        if (xmlHTTP.readyState == 4 && xmlHTTP.status == 200) {
            document.getElementById("SigningLink").innerHTML = '<a href="' +
xmlHTTP.responseText + '" target="_blank">' + 'Begin Signing' + '</a>';
            /* window.open(xmlHTTP.responseText);*/
        }
    }
    xmlHTTP.send(strRequest);
}
```

You can make the following changes in the CallAssureSign.js script to make it work with your manually designed eform.

- In the line **for (i = 0; i < row.length; i++)**, establish the iteration mechanism to implement proper traversal for your eForm.
- In the **var field = currentItem.childNodes.item(0).innerText;** line, under the **field** variable, assign the variable name of AssureSign parameter.
- In **the var dataItem = currentItem.childNodes.item(1).childNodes.item(0);** line, under the **dataItem** variable, assign the value of the AssureSign parameter that is identified in the **field** variable.

The strRequest = strRequest + "&" + field + "=" + value; line is eForm specific. This line denotes the relationship between AssureSign parameter names and associated values. The strRequest variable must be of the format ImageNowFormName=<Name of the Perceptive Content form>&parameter1=value1&parameter2 =value2 and so on, where parameter1 and parameter2 are AssureSign parameters and value1 and value2 are associated values.

The following URL in the above code snippet must point to the eForm service running as an eAuthorize component on PCR.

```
xmlHTTP.open("POST", "http://<Connect Runtime
host>:<port>/rs/SignatureEndpoint/SubmitToAssureSign", true)
```

Disable the Begin signing link

To disable the immediate presentment link, complete the following steps.

1. In the XSL file, delete ** ** in the body tag.
2. In the **CallAssureSign.js**, delete the following lines.

```
document.getElementById("SigningLink").innerHTML = 'Please wait....The document is
being submitted....Your signing link will appear here shortly';
and
   xmlHTTP.onreadystatechange = function () {
        if (xmlHTTP.readyState == 4 && xmlHTTP.status == 200) {
            document.getElementById("SigningLink").innerHTML = '<a href="' +
xmlHTTP.responseText + '" target="_blank">' + 'Begin Signing' + '</a>';
            /* window.open(xmlHTTP.responseText);*/
        }
    }
```

3. Restart **Tomcat** containing the forms server.

Disable buttons on eForms

Perceptive Forms Server provides the option to disable the save, print, reset, and attachments buttons.

1. Access the **$<IMAGENOWDIR6>\etc** directory and open the imagenowforms.xml file in a text editor.
2. Under the section where the **<FormName>** tag value is the name of the form you are using to sign.
3. Within **<ConfigParams>**, add the configuration parameters and set the values as **FALSE** for the buttons which you want to disable, as shown in the following example.

```
<DocumentForm>
    <FormName>eAuthorize_eForm</FormName>
    <QueueName></QueueName>
    <Drawer>Accounts Payable</Drawer>
    <Field1 isUnique="true"/>
    <Field2 isTimeStamp="true"/>
    <Field3/>
    <Field4/>
    <Field5/>
<ConfigParams>
<ConfigParam name=”saveVisible” value=”FALSE”/>
<ConfigParam name=”printVisible” value=”FALSE”/>
<ConfigParam name=”resetVisible” value=”FALSE”/>
   <ConfigParam name=”attachmetnsVisible” value=”FALSE”/>
</ConfigParams>
</DocumentForm>
```

4. Stop the **Engine** service and then restart **Tomcat**. The buttons are no longer displaying.

   Note If you save the attachments in Perceptive Content, you cannot submit them along with the signed document. As in this case, where you are not saving the form to Perceptive Content and are only collecting values from the eForm, you cannot submit attachments.

Map eForm fields to JotBlocks in AssureSign

To map the fields in forms with the parameters in JotBlock, consider the following points while creating eForms for submitting documents for signature.

- Create a document template in AssureSign matching the name of the form used for signature.

- JotBlocks should have **Type** as **Text**, **Input** as **Parameter** and the parameter name should match the field names in the form.
- **Workflow** template in AssureSign should contain the **Signatory Name** and **Email Address** matching those in the form.
- AssureSign treats the documents submitted for signature though eForm as third party documents. Hence, while creating **Workflow** template, on **Document Transmission**, select **eAuthorizeExternalDocUpload** as the **Design Name**.

Note The sample CallAssureSign.js provided with the installer is not a universal one. You may need to modify it in accordance with the form inputs.

Issues and workaround

Troubleshoot if bundles and components are not in Active state

Issue: All Bundles and Components mentioned in Verify the connector JAR bundle status section are not active.

Solution: This situation can occur if the connectors or related dependencies are either not installed or configured correctly in the Perceptive Connect Runtime. To solve this issue, perform the following steps.

1. Navigate to **Perceptive Connect Runtime** dashboard.
2. Configure and check that the configuration mentioned in the **Configure the Content Connector** section is correct and the bundles or components related to the Content Connector are in Active state.
3. In the **Perceptive Connect Runtime** dashboard, configure and check that the configuration mentioned in the **Configure the app inPerceptive Connect Runtime** is correct.
4. Restart the **Perceptive Connect Runtime** service.

Troubleshoot if the WSDL does not show the operations during Envoy creation

Issue: While configuring Envoy, the WSDL at **http://<Connect Runtime host>:<port>/ws/SignatureEndpoint?wsdl** does not show the relevant operations in the XML that appears.

Solution: This situation occurs if the connectors are not installed or configured correctly in Perceptive Connect Runtime. To solve this issue, perform the following steps.

1. Navigate to **Perceptive Connect Runtime** dashboard.
2. Configure and check that the configuration mentioned in the **Configure the app inPerceptive Connect Runtime** is correct.
3. In **Perceptive Connect Runtime** dashboard, check that all the bundles and components mentioned in the Verify the connector JAR bundle status section are in Active status.
4. Restart the **Perceptive Connect Runtime** service.

Troubleshoot if AssureSign is unable to send notifications to eAuthorize

Issue: AssureSign is unable to send notifications to eAuthorize

Solution: This situation occurs if the notifications configured in AssureSign are incorrect. To solve this issue, navigate to **AssureSign** administration page and verify the following settings:

- The notifications point to the correct Perceptive Connect Runtime service.
- The Request XML and Response XML of the notifications are correct as per **Set up the AssureSign environment** section.

Troubleshoot if there is Perceptive Connect Runtime startup failure

Issue: If Integration Server and Perceptive Forms Server are installed on the same Tomcat server, and one of the servers may fail to start causing Perceptive Connect Runtime startup failure.

Solution:

1. Cut **encryption.jar** from the **Tomcat\webapps\integrationserver\WEB-INF\lib** directory and paste to the **tomcat\webapps\shared** folder. Create the directory if it does not exist.
2. Open the **Tomcat\conf\catalina.properties** file and edit the shared.loader setting to **catalina_home\webapps\shared**.

Troubleshoot if Template ID is missing during eAuthorize document submission

Issue: The eAuthorize document submission fails and error logs display a message.

Solution: This error occurs if the AssureSign template is not properly configured. When you submit a document in Perceptive Content, the Document Type name in Perceptive Content must match with an AssureSign template name. The name match is case-sensitive. To know how to set up an AssureSign template, see the Set up an AssureSign template for signature section.

Troubleshoot if there is an installation issue after running the installer

Issue: After running the installer, installation issue may arise due to some reason.

Solution: If there is an installation problem, it is recommended that you must verify all the manual installation steps. For details, see the Install eAuthorize manually section.

Troubleshoot eForm

Issue: There may be an issue with the eForm.

Solution: If there is an issue with the eForm, complete the following steps.

1. Open the browser debug console and ensure that the **CallAsssureSign.js** file is properly loaded.

   Note If there is an error during loading the script file, ensure that the script file **.js** is included in the Presentation files list.
2. Restart the application server where the Forms server is hosted.
3. Add **console.log(strRequest)** in the script file after the entire request string is generated.

4. Save the script and refresh the browser to reload the form.
5. Enter the values in the input fields and click **Submit**.
6. Open the browser debug console and check the value of the strRequest variable appearing in the console which must be in the format **ImageNowFormName=<Name of the Perceptive Content form>&parameter1=value1&parameter2=value2**.

Note Ensure that there is no unnecessary space or unwanted charactersin the query parameter names. You must have a template with the same name as the eForm name configured in the AssureSign server and associated with a correct document transmission that points to the appropriate PCR server.

Appendix A: Manually create Perceptive Content components

Create new custom properties

See documentation to create new custom properties.

The following string properties are the input parameters for AssureSign Connector that you must provide:

- Signatory 1 Full Name
- Signatory 1 Email Address

The following string properties are output parameters for AssureSign Connector that automatically populate:

- AS_ID
- AS_SIGNATURE_STATUS
- AS_AUTH_TOKEN

The following Flag should be created

- AS_KEEP_ORIGINAL_DOCUMENT

The following List property should be created. Various actions occur depending on the value:

- AS_FAILURE_NOTIFICATION_TYPE
  - **Email**. If the document fails in the workflow, an email is sent to the email ID that you configured in Perceptive Content while creating user profiles. You must provide the SMTP server name, port, email ID in the **Configuration** page of **Perceptive Connect Runtime** dashboard, and the authentication if needed.
  - **Task**. You receive a task in **My Assigned** view in Perceptive Content that shows that the document failed in workflow.
  - **Both**. You receive an email in your email account and a task in Perceptive Content.

Create workflow for sending documents for signature

If you install eAuthorize manually, you must create a workflow process to send documents to AssureSign for signature and download the signed documents in Perceptive Content. These instructions may also be helpful if you want to modify the workflow that is automatically created by the install script.

You can only complete this procedure if you are a user with the global privilege to manage workflow processes, a manager, or the owner. This procedure creates automated system queues (ConnectQs).

To create your workflow process, complete the following steps.

1. On the **Perceptive Content** toolbar, click **Manage**, and then click **Workflow**.
2. On the **Workflow** tab, click **New**.
3. Enter a name for the workflow process.
4. Optional. Enter a description for your workflow process.
5. Click **OK**.
6. Double-click the process to open it in the **Workflow Designer**.
7. Create your workflow process by performing the following substeps.
    1. In the **Workflow Designer** window, in the left pane, under **Queues**, select the **Connect Queue** icon and drag it to the right in your workflow diagram. Repeat this step to create two more **Connect Queues**.

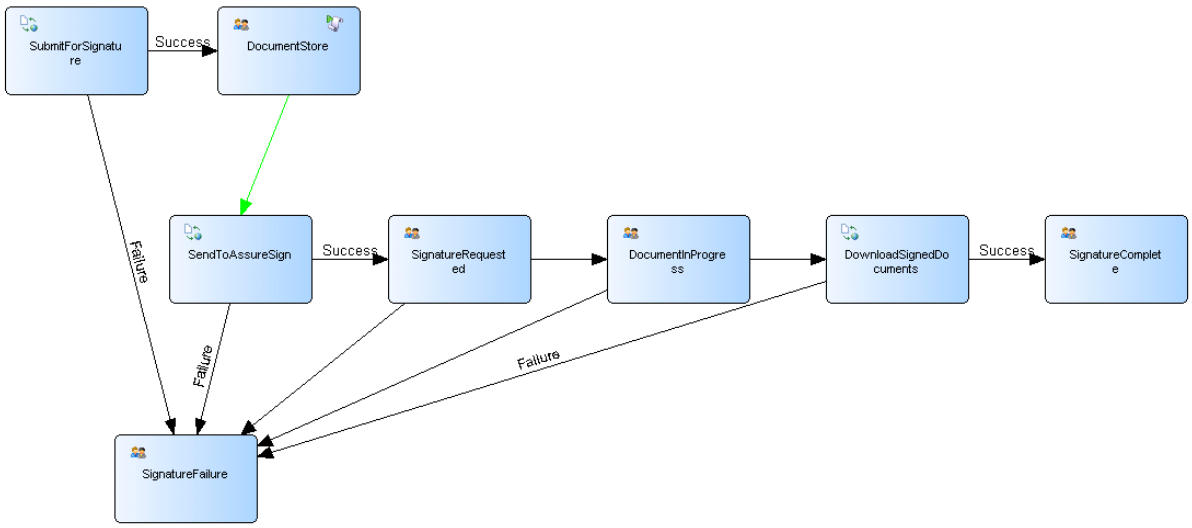
        Note In this example, the ConnectQs are SubmitForSignature, DownloadSignedDocuments, and SendToAssureSign. You can supply other names for the ConnectQs. To name each queue, double-click the queue, enter a name and then click **OK**.
    2. In the left pane, under **Queues**, select the **Work** queue and then drag it to the right in your workflow diagram.
    3. Double-click the queue to modify its properties.
    4. In the **Queue Properties** dialog box, in the left pane, select **Properties**.
    5. In the right pane, in the **Name** field, enter **SignatureRequested**.
    6. Click **OK**.
    7. Repeat the substeps 2 to 6 to create the **DocumentStore**, **DocumentInProgress**, **SignatureComplete**, and **SignatureFailure** work queues.
    8. Double-click **DocumentStore**.
    9. In the left pane, click **Actions** and complete the following substeps.
        1. Under **Inbound**, in the **iScript** list, select **Select iScripts** and click **Add**.
        2. Select **DocumentStoreAndForward.js** and click **OK**.
        3. Under **Inbound**, in the **iScript** list, select **DocumentStoreAndForward.js**.
    10. Click **OK**.
    11. Double-click **SignatureFailure**.
    12. In the left pane, click **Actions** and complete the following substeps.
        1. Under **Inbound**, in the **iScript** list, select **Select iScripts**, and click **Add**.
        2. Select **CreateTask.js** and click **OK**.
        3. Under **Inbound**, in the **iScript** list, select **CreateTask.js**.
    13. Click **OK**.

14. To configure **SubmitForSignature**, **DownloadSignedDocuments**, and **SendToAssureSign** Connect Queues, complete the followng substeps.

    Note If an older version of eAuthorize is already installed, delete the **SubmitToSignature**, **SendToAssureSign** and **DownlaodSigned Documents** Integration Connect Queues from the workflow. Currently, Integration ASQ is replaced with Connect Queues.

    1. Double-click the **SubmitForSignature** queue to modify its properties.
    2. Under **Automated Action**, configure the following options.
        - To designate the process and queue for items processed in **SubmitForSignature** queue, in the **Success Action** list, select the workflow process in the **Process** list, and select **DocumentStore** queue in the **Queue** list.
        - To designate the process and queue for items that do not successfully process in this queue, in the **Failure Action** list, select the workflow process created in the **Process** list, and select **SignatureFailure** queue in the **Queue** list.
        - To set the number of days that items remain in this queue after the business application receives a successful call for those items, in the **Route After (Days)** field, type a number. The maximum number of days is 365. By default, Perceptive Content routes items to the failure queue after one day.

          Note If your business application does not place a web service call for an item, that item remains in the Connect Queue for the number of days you specify in the **Route After (Days)** field.
    3. Repeat the steps above to configure **DownloadSignedDocuments** Connect Queue with the following configurations.
        - For **SuccessAction**, select **SignatureComplete** from **Queue** list, and for **FailureAction**, select **SignatureFailure** from the **Queue** list.
    4. Repeat the steps above to configure **SendToAssureSign** Connect Queue with the following configurations.
        - For **Success Action**, select **SignatureRequested** from **Queue** list, and for **Failure Action**, select **SignatureFailur**e from the **Queue** list.
    5. To create a complete workflow, join the queues as shown in the following figure.



6. To create the routes between the queues, complete the following substeps.

    1. In **Workflow Designer**, in the **Task** pane, click **Routes**.
    2. On the **Grid** toolbar, verify that the **Normal Routes** button is selected.
    3. Under **Routes**, select **Sequential route**.
    4. To create a route from one queue to another queue, click on the queue where you want the route to begin and drag your cursor to the queue where you want the route to end.

    Note Ensure that the route from **DocumentStore** to **SendToAssureSign** is a **Seq-Auto route**.

    5. After completing the workflow process creation, close the **Workflow Designer**.

Note If the flow of documents is interrupted anywhere in this workflow it redirects to the **SignatureFailure** queue. To route the documents manually to the queue from where it was interrupted, you can open the document in **Workflow** and click **RouteBack**.

Create task templates

You receive a task in the My Assigned view in Perceptive Content when a submitted document fails in workflow. You can also opt to receive email notifications when submitting the document for signature.

The configuration iScript, ConfigurePerceptiveContentForEAuthorize.js, automatically creates task templates, reasons, and a reasons list. The script also populates the reason lists with the corresponding reason list member and associates the appropriate action reasons and return reasons with each task template. For a table showing these correlations, see the Task templates section.

To configure the task templates, complete the following steps.

1. On the **Perceptive Content** toolbar, click **Manage**.
2. In **Management Console**, in the left pane, click **Tasks**.
3. In the right pane, on the **Templates** tab, in the **Select a task type** field, click **Pointer**.
4. On the **Templates** tab, select **FailureNotification_Cancelled** and click **Modify**.
5. In the **Pointer Task** dialog box, complete the following steps.

1. In the left pane, select **Properties**.
    1. In the **Description** field, type a template description.
    2. Under **Options**, check the **Is active** check field to make the task template available to task creators assigning tasks from the **Tasks** toolbar.
2. In the left pane, click **Components**.
    1. In the right pane, under **General**, in the **Task instructions** field, type the instructions you want your task assignees to see. These instructions are the content for the email notifications that the system sends to the task assignee. You can base your instructions on the **Action Reason List** column in the table in the Task templates section.
    2. If you want to allow a task creator to modify the instructions on the **Options** tab in the **NewTask** dialog box, ensure that the **Modifiable during task creation** check box is selected.
    3. In the **Task location** section, select the following locations to determine where tasks created with this template are assigned.
        - **Folder.** Create a task for a folder.
        - **Document.** Create a task for a document.
        - **Page without a visual representation.** Create a task for a page in a document with no visual representation.
        - **Page with a visual representation.** Create a task, along with a visual representation, for a page in a document.
    4. In the **Completion** field, select **Manual**.
    5. Optional. On **WorkflowAssignment**, in the **Send to queue** list, select **(None)**.
3. In the left pane, click **Assignment**.
    1. In the right pane, click **Add**.
    2. In the **Select Users and Groups** dialog box, assign the users and groups you want to have access to this task.
4. Optional. In the left pane, click **Reasons** and then select the **Assignee must specify a reason during task completion** check box to require task assignees to select a reason after completing a task.

6. Repeat the procedure four more times to configure the remaining eAuthorize pointer task templates.
    - FailureNotification_Declined
    - FailureNotification_DownloadFailed
    - FailureNotification_Expired
    - FailureNotification_SubmissionFailed

Create Audit History form

You can complete this procedure only if you are an owner or manager, or are assigned the Manage Forms privilege. A Forms license is required to use forms in eAuthorize. Without this license, the audit trail history will not be available in Perceptive Content but will be available from AssureSign. If you need a Forms license, contact your Perceptive Software representative.

This form is created as part of the **Run Configure script to create the default components** step. However the form can be manually installed.

- The form is required to be named with the words “AssureSign”, “Audit”, and “History”. This requirement is not case-sensitive.
- It is recommended that you create only one Audit History form, because data is populated in the form which appears first in the list of forms, alphabetically. If you erroneously create more forms, delete them.

Find the provided files in the <**EAUTHORIZEINSTALLDIR**>/eAuthorize/forms/AuditHistory. This folder contains two folders, data_definition and presentation.

- The data definition for AssureSign Audit History form is available in the **data_definition** folder.
- The XSL file and other supporting files required for the presentation of this form are available in the **presentation** folder.
- See Documentation for details on how to **Manage manual forms > Manually modify form components**

The Audit History form has a refresh timeout configuration option, you can set this option in the $<IMAGENOW6DIR>/etc/eAuthorize/eAuthorize_config.xml.To configure this option, provide the time out in seconds within the **<refreshTimeout>** tag.

Create ImmediatePresentment form

You can complete this procedure only if you are an owner or manager, or have the Manage Forms privilege. A Forms license is required to use forms in eAuthorize. If you need a Forms license, contact your Perceptive Software representative. If the form license is not activated, the AssureSign ImmediatePresentment form is not available in Perceptive Content.

This form is created as part of the **Run Configure script to create the default components** step. However the form can be manually installed.

- The form is required to be named with the words “AssureSign”, “Immediate”, and “Presentment”. This requirement is not case-sensitive.
- It is recommended that you create only one AssureSign ImmediatePresentment form, because data is populated in the form which appears first in the list of forms, alphabetically. If you have erroneously created more forms, delete them.

Find the provided files in the <**EAUTHORIZEINSTALLDIR**>/eAuthorize/forms/ImmediatePresentmentForm. This folder will contain two folders, data_definition and presentation.

- The data definition for AssureSign ImmediatePresentment form is available in the **data_definition** folder.
- The XSL file and other supporting files required for the presentation of this form are available in the **presentation** folder.
- See Documentation for details on how to **Manage manual forms > Manually modify form components**

Appendix B: About the eAuthorize Perceptive Content configuration

The eAuthorize solution uses various products for the signing and storage of documents. This section of the installation guide describes the Perceptive Content configuration provided with the ConfigurePerceptiveContentForEAuthorize.js and CreateWorkflowForEAuthorize.js configuration iScripts. After you run the scripts, the system creates the storage and workflow needed by eAuthorize on your Perceptive Content system.

This appendix provides details about the Perceptive Content configuration that is automatically set up by the configuration iScripts.

Custom properties

The following table provides the list of custom properties that the configuration iScripts create during installation.

Custom Property Name	Description	Parameter Type	Data Type
AS_AUTH_TOKEN	Unique security token from AssureSign	Output parameter	String
AS_FAILURE_NOTIFICATION_TYPE	Type of notification sent to the user when a document fails in workflow	Input parameter	List
AS_ID	Unique ID from AssureSign	Output parameter	String
AS_KEEP_ORIGINAL_DOCUMENT	If this is set as TRUE, the original document remains as is in the DocumentStore work queue in Perceptive Content workflow and a copy of it is created.	Input parameter	Flag
AS_SIGNATURE_STATUS	Status of the signature in workflow	Output parameter	String
Signatory 1 Full Name	Name of the person who needs to sign the document	Input parameter	String
Signatory 1 Email Address	Email address of the person who needs to sign the document	Input parameter	String
Signatory 2 Full Name	Name of the person who needs to sign the document	Input parameter	String
Signatory 2 Email Address	Email address of the person who needs to sign the document	Input parameter	String
Signatory 3 Full Name	Name of the person who needs to sign the document	Input parameter	String
Signatory 3 Email Address	Email address of the person who needs to sign the document	Input parameter	String

It is important to understand how these custom properties function.

- Signatory 1 Full Name and Signatory 1 Email Address are the input parameters for AssureSign Connector that you must provide.
- AS_ ID, AS_AUTH_TOKEN, and AS_SIGNATURE_STATUS are the output parameters for AssureSign Connector that populate automatically.
- Various actions occur, depending on the value you set for AS_FAILURE_NOTIFICATION_TYPE:
    - **Email**. If the document fails in the workflow, an email is sent to the email ID that you configured in Perceptive Content while creating user profiles. You must provide the SMTP server name, port, email ID in the **Configruation** page of **Perceptive Connect Runtime** dashboard, and the authentication if needed.
    - **Task**. You receive a task in the My Assigned view in Perceptive Content that shows that the document failed in workflow.
    - **Both**. You receive an email in your email account and a task in Perceptive Content.

About custom properties for extended features

To send a document to multiple signatories, you need the signatory full name and signatory email address custom properties corresponding to each signatory. During installation, the following custom properties are automatically created for sending a document to additional signatories.

- Signatory 2 Full Name
- Signatory 2 Email Address
- Signatory 3 Full Name
- Signatory 3 Email Address

When you create the multiple signatories, you must understand how the information correlates to the AssureSign template and Perceptive Content custom properties.

- You specify the signatory names and email addresses while creating a template in AssureSign.
- You must make sure that the custom properties for signatory full name and signatory email address in Perceptive Content match the AssureSign template. For example, if you use Signatory 1 Full Name and Signatory 1 Email Address in a particular template, make sure that Perceptive Content uses the same Signatory 1 Full Name and Signatory 1 Email Address custom properties.

Note Additional custom properties can be used to prefill information on an AssureSign document if the name of AssureSign template parameter and the names of custom property are same.

For details, see the AssureSign Quick Reference Guide for instructions on creating a template. Ensure that each name and email address corresponds to custom properties in Perceptive Content.

Task templates

The configuration iScript, ConfigurePerceptiveContentForEauthorize.js, automatically creates pointer task templates, reasons, and reason lists. The iScript also populates the reason lists with the corresponding reason member and associates the appropriate action reasons and return reasons with each task template. The following table shows these correlations.

Task Name	Action Reason List	Action Reason List Members	Return Reason List	Return Reason List Members
FailureNotification_Cancelled	Cancelled	Document/Envelope is cancelled by signatory.	Task Return List	Additional information requested. Not my document. Not my folder. Not completed as required. See comments for more information.
FailureNotification_Declined	Declined	Document/Envelope is declined by signatory.	Task Return List	Additional information requested. Not my document. Not my folder. Not completed as required. See comments for more information.
FailureNotification_DownloadFailed	DownloadFailed	Document/Envelope download failed.	Task Return List	Additional information requested. Not my document. Not my folder. Not completed as required. See comments for more information.
FailureNotification_Expired	Expired	Document/Envelope is expired.	Task Return List	Additional information requested. Not my document. Not my folder. Not completed as required. See comments for more information.
FailureNotification_SubmissionFailed	Submission Failed	Document/Envelope submission failed.	Task Return List	Additional information requested. Not my document. Not my folder. Not completed as required. See comments for more information.

eAuthorize iScripts

The following iScripts are in the [*drive*:]\inserver6\script directory.

Script	Function
CreateTask.js	For creating a task in Perceptive Content if a document fails in workflow.
DocumentStoreAndForward.js	For storing and forwarding a document in a workflow queue.
RefreshAuditHistory.js	For refreshing Audit History.

The following files are in the [*drive*:]\inserver6\script\eAuthorize directory.

- IN_WorksheetManager.jsh
- IN_XML.jsh
- INBasePath.jsh
- Util_Misc.jsh

Note For more information of these files, refer to the **Install Perceptive Content files** section.

AssureSign forms

A Forms license is required to use forms in eAuthorize. If you need a Forms license, contact your Perceptive Software representative.

AssureSign Audit History form

This form displays the audit trail of each document processed in eAuthorize. If the form license is not activated, the audit trail history will not be available in Perceptive Content but will be available in AssureSign.

About the AssureSign Audit History form files and components

The configuration iScript, ConfigurePerceptiveContentForEauthorize.js, automatically creates the AssureSign Audit History form for you. You create a form by first uploading the data definition file and then the XSL style sheet and supporting files. The iScript also automatically loads these files to the [*drive*:]\inserver6\forms directory and configures the corresponding Perceptive Content components. To create it manually, see the Set up eForm for AssureSign Audit History section.

File name	Purpose
AssureSign_AuditHistory.xml	The data definition XML file contains the schema used to save data instances in data content record files.
AssureSign_Audit_History_Presentation.xsl	The presentation XSL file describes how to present the XML data in the form.
Supporting files	The supporting files for the presentation. • AssureSign_AuditHistory.xls • Asynch.js • Forms.css • Refresh_16.png • Section_header_bg.gif

AssureSign ImmediatePresentment form

This form provides you the link to sign the associated documents from within Perceptive Content client. If the form license is not activated, the AssureSign ImmediatePresentment form is not available in Perceptive Content.

About AssureSign ImmediatePresentment form files and components

The configuration iScript, ConfigurePerceptiveContentForEauthoriz.js, automatically creates the AssureSign ImmediatePresentment form for you. You create a form by first uploading the data definition file and then the XSL style sheet and supporting files. The iScript also automatically loads these files to the [*drive:*]\inserver6\forms directory and configures the corresponding Perceptive Content components. To create this manually, see the Set up eForm for AssureSign ImmediatePresentment.

File name	Purpose
AssureSign_ImmediatePresentment.xml	The data definition XML file contains the schema used to save data instances in data content record files.
AssureSign_ImmediatePresentment.xsl	The presentation XSL file describes how to present the XML data in the form.
Supporting files	The supporting files for the presentation. • Forms.css • Forms.js • section_header_bg.gif

About the workflow for sending documents for signature

A workflow process is required to send documents to AssureSign for signature and download the signed documents in Perceptive Content. The installer automatically configures workflow configurations.

The configuration iScript, CreateWorkflowForEAuthorize.js, **c**reates a workflow process and the queues described in the following table.

Queue Name	Description
SubmitForSignature	Connect Queue for submitting a document for signature. This is the first queue in the workflow process.
DocumentStore	When AS_KEEP_ORIGINAL_DOCUMENT is TRUE, the original document remains as is in this work queue, a copy of it is created in Perceptive Content, and through sequential auto-routing it is forwarded to the Perceptive Content ASQ.
SendToAssureSign	When the document is submitted to AssureSign for signature.
SignatureRequested	When the email notification from AssureSign is sent to signatory.
DocumentProgress	When the signing process is in progress.

Queue Name	Description
DownloadSignedDocuments	Connect Queue where the signed document is routed for downloading in Perceptive Content.
SignatureComplete	Work queue where the document is routed when the required signature is obtained.
SignatureFailure	Cancelled, expired, declined, or format mismatched documents are routed to this queue.

Appendix C: Configuration

The following table provides definitions and instructions for setting the parameter values in the Configuration page of Perceptive Connect Runtime dashboard.

Section	Parameter	Guideline
AssureSign parameters configuration	WSDL_URL	A URL that defines the location of the sandbox or production instance in the AssureSign cloud, or a URL to the local AssureSign web application you are using.
	Local domain name	Keep the field blank if you are using an instance of the AssureSign cloud, for example: <parameter name= "AS_LOCAL_INSTANCE_DOMAIN_NAME"value=""/ > If you are using a local AssureSign web application, provide the machine name as the value. You can also find the machine name in self-signed certificate issued during AssureSign installation.
	User name	An AssureSign administrative account that is used to authenticate into AssureSign and submit documents for signature.
	Context ID	AssureSign's DocumentNOW Account Context Identifier specifies a unique identifier needed in order to validate the request. You can find this identifier in the **Settings** section of the Administrative tab within your AssureSign environment, if you have administrative access.
	Template tag	This value needs to match the Template tag value in the AssureSign template. Use the value that you have already entered in AssureSign, or note this value so that you can enter it in AssureSign as the Template Tag. For details, see the AssureSign Quick Reference Guide for instructions to create a template.

Section	Parameter	Guideline
Email parameters configuration	SMTP server name	The name of the SMTP server that sends email notifications.
	SMTP server port	The port of the SMTP server that sends email notifications.
	Sender email ID	The email ID configured on the SMTP server that sends email notifications.
	SMTP Authentication	If your SMTP email server requires authentication, provide the value as ‘true’; otherwise, use "false".
	Authentication user name	If your SMTP email server requires authentication, provide a username for the email server.
	Authentication password	If your SMTP email server requires authentication, provide the password corresponding to above-mentioned username for the email server.
	Start TLS	If your SMTP email server requires TLS encryption, provide the value as ‘true’; otherwise, use "false".
	Encryption Protocols	If your SMTP email server requires encryption protocols, provide a space separated list of supported encryption protocols to negotiate when connecting to the email server. Example protocols: ‘SSLv3 TLSv1 TLSv1.1 TLSv1.2’
	Check Server Identity	If your SMTP server has a signed certificate, provide the value as ‘true’ to validate the server’s identity; otherwise, use “false”. SMTP servers with self-signed or unsigned certificates must use “false”.

Appendix D: Copy web notification templates

If you are using a local instance of the AssureSign environment, complete the following steps to copy the web notification templates from the sandbox environment.

1. In a browser window, sign into the **AssureSign sandbox environment**.
2. In another browser window, sign into your **AssureSign local environment**.
3. In the **sandbox** environment, on the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration.**
4. In your **local** environment, on the **Administration** tab, click **DocumentTRAK** and then click **Notification Administration.**
5. In the **sandbox** environment, in the **WEBHOOKS** table, locate the design you want to copy and click **Copy**.

6. In your **local** environment, in the **WEBHOOKS** table, click **ADD WEBHOOK** and then select **Custom**.
7. In the **sandbox** environment, copy the values from the **General Information** page to the **General Information** window in your local environment.
8. In your **local** environment, in the **General Information** page, click **Continue** two times.
9. Click the ellipsis button and then change **Content-Type** to **text/xml**.
10. Click **Save Header** and then click **Continue.**
11. To copy the request XML from the sandbox environment, complete the following substeps.
    1. In the **sandbox** environment, in the **General Information** page, click **Continue** three times.
    2. In the **Request Body** page, copy the XML.
    3. In your **local** environment, paste the XML. (Replace any XML that was already there.)
    4. In both the **sandbox** and **local** environments, click **Continue** twic**e**.
12. To copy the response validation method from the sandbox environment, complete the following substeps.
    1. In the **sandbox** environment, click the **Edit validator** ellipsis button and then opy all field values.
    2. In your **local** environment, click the **+ button** and then paste all field values.
    3. In both the **sandbox** and **local** environments, click **Finsh**.
13. Repeat the above steps for each required template.