



## Integration for Adobe Sign

### Reference Guide

---

*Includes:*

Installation Guide

Administration Guide

User Guide

# Copyright

Information in this document is subject to change without notice. The software described in this document is furnished only under a separate license agreement and may be used or copied only according to the terms of such agreement. It is against the law to copy the software except as specifically allowed in the license agreement. This document or accompanying materials contains certain information which is confidential information of Hyland Software, Inc. and its affiliates, and which is subject to the confidentiality provisions agreed to by you.

All data, names, and formats used in this document's examples are fictitious unless noted otherwise. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright law, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Hyland Software, Inc. or one of its affiliates.

Hyland®, Hyland Software®, Hyland Healthcare, and Hyland product names are registered and/or unregistered trademarks of Hyland Software, Inc. and its affiliates in the United States and other countries. All other trademarks, service marks, trade names and products of other companies are the property of their respective owners.

© 2021 Hyland Software, Inc. and its affiliates. All rights reserved.

Document Name .....Integration for Adobe Sign  
Department/Group ..... Documentation  
Revision Number .....Foundation EP5

**OVERVIEW**

|                        |          |
|------------------------|----------|
| <b>Licensing .....</b> | <b>1</b> |
|------------------------|----------|

**INSTALLATION**

|   |           |
|---|-----------|
| <b>Requirements .....</b>   | <b>3</b>  |
| General Requirements .....  | 3         |
| .NET Core Requirements .....  | 3         |
| Third-Party Software .....  | 3         |
| <b>Licensing .....</b>  | <b>3</b>  |
| <b>Pre-Installation .....</b>   | <b>4</b>  |
| <b>Installation .....</b>   | <b>4</b>  |
| Installing the Hyland Electronic Integration Service .....                        | 4         |
| Running the Hyland Electronic Signature Service Database Configuration Tool ..... | 8         |
| Oracle Environments .....   | 8         |
| SQL Environments .....  | 9         |
| <b>Troubleshooting .....</b>  | <b>10</b> |
| Cannot Connect OnBase to Adobe Sign .....   | 10        |
| Encryption Prevents Connection to Adobe Sign .....                                | 10        |
| Cannot Enter IdP Credentials When Connecting to Adobe Sign .....                  | 10        |
| DNS or 404 Errors Displayed After Connecting to Adobe Sign .....                  | 11        |
| Cannot Configure Refresh Token .....  | 11        |
| Cannot Execute Adobe Sign Workflow Rules or Actions.....                          | 11        |
| Cannot Connect to a Data Source.....  | 12        |
| Cannot Create Trust Configuration.....  | 12        |
| Cannot Send or Retrieve Adobe Sign Documents.....                                 | 12        |
| <b>Contacting Support .....</b>   | <b>13</b> |

**CONFIGURATION**

|   |           |
|---|-----------|
| <b>Pre-Configuration .....</b>                    | <b>15</b> |
| Configuring Document Types for Signing .....      | 15        |
| Configuring Document Type Auto-Name Strings ..... | 15        |
| Receiving Email Notifications.....                | 16        |
| <b>Connecting Adobe Sign to OnBase .....</b>      | <b>16</b> |
| <b>System Interaction .....</b>                   | <b>18</b> |
| Workflow .....                                    | 18        |
| Upload to Adobe Sign Action .....                 | 18        |
| Retrieve from Adobe Sign Action .....             | 21        |
| Check Agreement Status Rule .....                 | 21        |

**USAGE**

Integration for Adobe Sign provides complementary functionality to Adobe Sign by automating many of the tasks surrounding signature management. Adobe Sign is an eSignature service, where a packet (a batch of one or more documents) can be uploaded and signed securely and electronically online.

Documents to be signed can be uploaded to Adobe Sign by using a Adobe Sign-specific Workflow action (must be licensed for Workflow). E-mail notifications alert signers that a document is ready for them to sign, and once a document is signed by all required parties, it becomes read-only and can be returned to the OnBase system.

## Licensing

In addition to a base package license for standard OnBase functionality, the **Integration for Adobe Sign** license is required to access standard Integration for Adobe Sign functionality.



# **Integration for Adobe Sign**

## **Installation Guide**

## Requirements

The following sections outline requirement information specific to Integration for Adobe Sign in OnBase Foundation EP5.

### General Requirements

For general requirement information that applies to Integration for Adobe Sign and other modules, see the sections on the following topics in the **Installation Requirements** manual:

- Database requirements
- Supported desktop operating systems
- Microsoft .NET Framework requirements
- Microsoft Visual C++ requirements
- Server and core workstation hardware requirements
- Web browser requirements

### .NET Core Requirements

The Integration for Adobe Sign requires the Microsoft .NET Core 6.0 Runtime and Hosting Bundle.

---

**Note:** The .NET Core Runtime and Hosting bundle must be installed prior to installing the Integration for Adobe Sign.

---

The .NET Core Runtime and Hosting Bundle is a Microsoft product. For installation and configuration procedures, see Microsoft's .NET Core documentation.

### Third-Party Software

The Integration for Adobe Sign requires an Adobe Sign solution.

## Licensing

See [Licensing on page 2](#) for licensing requirements.

## Pre-Installation

Before installing the Integration for Adobe Sign, you must install and configure the Hyland IdP, API Server (which includes the Hyland Electronic Integration Service), and Application Server. For more information on installing and configuring these modules, see the **Identity and Access Management Services** documentation, the **API Server** documentation, and the **Application Server** documentation.

---

**Note:** The IIS application pool user for the API Server must have access to IIS resources and permissions to write to the temporary files folder.

---

In addition, you must configure an Adobe API Application for use with the Integration for Adobe Sign. When configuring OAuth for the Adobe API Application, consider the following:

- You must configure the Adobe API Application redirect URLs with information specific to your solution). For example:  
`https://machinename/apiserver/esign/esignature/auth/callback;https://machinename/apiserver/esign/esignature`
- The following scopes are required at minimum:
  - **user\_read**
  - **user\_login**
  - **agreement\_read**
  - **agreement\_write**

See Adobe Sign documentation for information on creating an Adobe API Application.

## Installation

To install the Integration for Adobe Sign, you must complete these tasks:

- Install the Hyland Electronic Integration Service. See [Installing the Hyland Electronic Integration Service on page 4](#) for more information.
- Run the Hyland Electronic Signature Service Database Configuration Tool to update your database with the required schemas and tables. See [Running the Hyland Electronic Signature Service Database Configuration Tool on page 8](#) for more information.

### Installing the Hyland Electronic Integration Service

To install the Hyland Electronic Integration Service, follow these steps:

1. Install the API Server to deploy the Hyland Electronic Integration Service. See the **API Server** documentation for more information.
2. Open the **1\_Server.json** file in a text editor such as Notepad.

---

**Note:** In Windows, the JSON file must be opened as an administrator.

---

3. Update the **Address** element (under **Authentication | Provider**) to be the address of your IDP server.

4. Save and close the **1\_Server.json** file.
5. Open the **5\_esignature.json** file in a text editor such as Notepad.

---

**Note:** In Windows, the JSON file must be opened as an administrator.

---

6. Update the following fields (under **ElectronicSignatureService | TenantList**) as required:

| Field                   | Description  |
|-------------------------|--|
| <b>DataSource</b>       | <p>The name of the data source for your OnBase solution. This must match the tenant name configured for your Hyland IdP.</p> <hr/> <p><b>Note:</b> This value must be unique.</p> <hr/>  |
| <b>DataProvider</b>     | <p>The database provider type. Set the value to <b>MSSQL</b> for Microsoft SQL or <b>ORACLE</b> for Oracle.</p>  |
| <b>ConnectionString</b> | <p>A valid connection string to the OnBase database for the database provider type (SQL or Oracle). The connection string must include <b>Data Source</b>, <b>database</b>, <b>User Id</b>, and <b>Password</b> information.</p> <hr/> <p><b>Tip:</b> ADO connections strings are a method of connecting applications to databases. Complete details on connection strings and how to create them are available from Microsoft.</p> <hr/> <p><b>Note:</b> Make sure you include the double-slash ( \ ) between the server and the database instance to account for JSON formatting. For example, <b>Data Source=MyDB\\SQLInstance;</b></p> <hr/> |

7. Configure encryption settings for your solution as required. You should configure your solution to use one of the following encryption methods:
  - **Certificate thumbprint encryption:** this is the recommended encryption method. This encryption method uses a certificate thumbprint. When using this encryption method, you must configure the **CertificateThumbPrint** setting in the **5\_esignature.json** file. In addition, it is considered a best practice to configure the **KeyDirectoryPath** setting in the **5\_esignature.json** file.
  - **DPAPI-NG encryption:** this encryption method uses DPAPI-NG. When using this encryption method, it is considered a best practice to configure the **ProtectionDescriptorRule** and **KeyDirectoryPath** settings in the **5\_esignature.json** file.

---

**Note:** DPAPI-NG encryption is only supported on Windows 8 or higher and Windows Server 2012.

---



8. Set the **EncryptionSettings** settings as required:

| Field                           | Description  |
|---------------------------------|--|
| <b>ProtectionDescriptorRule</b> | <p><b>Note:</b> This setting is not supported for use with certificate thumbprint encryption. You should only configure this setting if you are using DPAPI-NG encryption.</p> <p>Set to the SID of an active directory group. The SID must be set using the following format: "SID=MySIDNumber", where <b>MySIDNumber</b> is replaced with the SID number to be used. For example:</p> <pre>"ProtectionDescriptorRule": "SID=S-1-1-12-12345678-123456789-123456789-12345"</pre> <p>When using this setting, it is recommended to create a shared directory used to store the encryption key. This directory can then be entered as the <b>KeyDirectoryPath</b> later in this procedure. If setting up multiple instances of the Hyland Electronic Integration Service, it is recommended that the <b>KeyDirectoryPath</b> be a shared directory that each instance can access.</p> <p><b>Note:</b> If this setting is left blank, the SID of the application pool identity will be used.</p> <p><b>Caution:</b> If this setting is left blank and the application pool user is not a user in your active directory domain, the Integration for Adobe Sign will not be able to use the key generated in <b>KeyDirectoryPath</b> to encrypt the data.</p> |

9. Set the **LocalStorageManagement** settings as required:

| Field                   | Description   |
|-------------------------|---|
| <b>KeyDirectoryPath</b> | <p><b>Note:</b> This setting is supported for use with certificate thumbprint encryption and DPAPI-NG encryption.</p> <p>Set to the directory used to store the encryption key created when you link your Adobe account to Studio. For example, <b>%LocalAppData%\MyKeysDirectory</b></p> <p>It is considered a best practice to manually create a directory and configure the <b>KeyDirectoryPath</b> setting for that directory.</p> <p>However, if this setting is left blank, a directory will be created automatically when you link your Adobe account to Studio, as long as the application pool user has access to the directory location being created. The service will attempt to create a directory in the following locations, in order:</p> <ul style="list-style-type: none"> <li>• %localAppData%\esignkeys</li> <li>• %AppData%\esignkeys</li> <li>• HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ASP.NET\4.0.30319.0\AutoGenKeys\{CurrentUserSID}\EsignKeys</li> <li>• the installation location of the API Server</li> </ul> <p><b>Caution:</b> If the application pool user does not have access to any of the above, no directory will be created and Adobe Sign credentials cannot be stored.</p> <p><b>Note:</b> If deploying in IIS, ensure that the application pool user has permission to access the directory used in <b>KeyDirectoryPath</b> setting as well as access to IIS resources and permissions to write to the temporary files folder.</p> |

| Field                        | Description   |
|------------------------------|---|
| <b>CertificateThumbPrint</b> | <p><b>Note:</b> This setting is not supported for use with DPAPI-NG encryption. You should only configure this setting if you are using certificate thumbprint encryption.</p> <hr/> <p>Set to the thumbprint for the certificate.</p> <p>This setting is required to use certificate thumbprint encryption, which is supported for both Windows and Linux environments. The user running the process must have access to the certificate.</p> <p>When using this setting, it is recommended to create a shared directory used to store the encryption key. This directory can then be entered as the <b>KeyDirectoryPath</b>. If setting up multiple instances of the Hyland Electronic Integration Service, it is recommended that the <b>KeyDirectoryPath</b> be a shared directory that each instance can access.</p> |

10. Save and close the **5\_signature.json** file.

## Running the Hyland Electronic Signature Service Database Configuration Tool

You must run the Hyland Electronic Signature Service Database Configuration Tool (also known as **Hyland-eSignature-Config**) to update your database with the required schemas and tables.

See one of the following sections for information on running this configuration tool for a SQL or Oracle environment:

- [Oracle Environments on page 8](#)
- [SQL Environments on page 9](#)

### Oracle Environments

1. A certified Oracle database administrator must create a new database user with administrative privileges. This user must be named **HSIESIGNATURE** (in all capital letters).
2. A certified Oracle database administrator must run the following script upon the database:

```
CREATE USER HSIESIGNATURE IDENTIFIED BY password;
GRANT UNLIMITED TABLESPACE TO HSIESIGNATURE;
GRANT CREATE SESSION TO HSIESIGNATURE;
GRANT CREATE TABLE TO HSIESIGNATURE;
GRANT SELECT_CATALOG_ROLE TO HSIESIGNATURE;
GRANT CREATE SEQUENCE TO HSIESIGNATURE;
```

**Note:** In the above script, **password** must be replaced by the password used by the **HSIESIGNATURE** user account.

3. Locate the Hyland Electronic Integration Service configuration tool package within your solution files. For instance, **hyland-eSign-config 0.1.0-alpha0489.nupkg**

---

**Tip:** The latest version of the **Hyland-eSignature-Config** configuration tool is available on Community at <https://community.hyland.com/customer-portal/5097/downloads>

---

4. Extract the package to a local directory.
5. Open the **hyland.esignatureservice.configuration.appsettings.json** file using a text editor such as Notepad.
6. Modify the **DataSource**, **DataProvider**, and **ConnectionString** fields so that they contain connection information for your database. If your solution uses a single OnBase database, it is recommended to use the same connection string for maintenance purposes.

---

**Caution:** You can only enter the information for a single database. Multiple tenants are not supported.

---

7. Save and close the **hyland.esignatureservice.configuration.appsettings.json** file.
8. Open a command prompt with administrative privileges.
9. Within the command prompt, navigate to the location of the directory where you extracted the Hyland Electronic Integration Service Database Configuration Tool package.
10. Run the following command:  
`dotnet Hyland.ESignatureService.Configuration.dll`
11. Close the command prompt.

## SQL Environments

1. Locate the Hyland Electronic Integration Service Database Configuration Tool package within your solution files. For instance, **hyland-eSign-config 0.1.0-alpha0489**

---

**Tip:** The latest version of the **Hyland-eSignature-Config** configuration tool is available on Community at <https://community.hyland.com/customer-portal/5097/downloads>

---

2. Extract the package to a local directory.
3. Open the **hyland.esignatureservice.configuration.appsettings.json** file using a text editor such as Notepad.
4. Modify the **DataSource**, **DataProvider**, and **ConnectionString** fields so that they contain connection information for your database. If your solution uses a single OnBase database, it is recommended to use the same connection string for maintenance purposes.

---

**Caution:** You can only enter the information for a single database. Multiple tenants are not supported.

---

5. Save and close the **hyland.esignatureservice.configuration.appsettings.json** file.
6. Open a command prompt with administrative privileges.

7. Within the command prompt, navigate to the location of the directory where you extracted the Hyland Electronic Integration Service Database Configuration Tool package.
8. Run the following command:  
`dotnet Hyland.ESignatureService.Configuration.dll`
9. Close the command prompt.

## Troubleshooting

The following sections describe common errors and how to resolve them.

### Cannot Connect OnBase to Adobe Sign

**Issue:** The following error message is displayed when attempting to connect to Adobe Sign from OnBase Studio:

**Unable to authorize access because the client configuration is invalid: invalid\_request**

**Resolution:** Ensure that Adobe Account information entered in Studio is correct. Remember that the **Electronic Signature Integration Service URL** field is case-sensitive, and the **Adobe Sign Hostname URL** field must be adjusted for your region.

This issue can also be caused if the redirect information is not correct for your Adobe Sign account, or if the Adobe Account API does not have OAuth set up. See your Adobe Sign documentation for information on configuring redirect information and OAuth protocol.

### Encryption Prevents Connection to Adobe Sign

**Issue:** The following error messages are displayed in the Diagnostics Console when attempting to connect to Adobe Sign:

**Please configure encryption settings LocalStorage management for key storage**

**Please check permissions on your service and/or configure encryption settings LocalStorage management for key storage**

**Resolution:** This issue occurs when OnBase is unable to store Adobe Sign credentials (for example, if the default DPAPI-NG encryption method is used but the application pool user is unable to access the directory used to store the encryption key. To resolve this issue, create a new directory to store the encryption credentials and ensure your application pool user has access to it. The directory must be configured in the **KeyDirectoryPath** setting.

### Cannot Enter IdP Credentials When Connecting to Adobe Sign

**Issue:** When entering your Hyland IdP credentials while connecting an Adobe Sign account to OnBase, the following error is displayed:

**Sorry, there was an error**

In addition, the diagnostic console displays one of the following errors:

- **Internal.Cryptography.CryptoThrowHelper+WindowsCryptographicException: Keyset does not exist**  
**Resolution:** Grant the IdP application pool read access to the certificate being used for signing/encryption. See Microsoft's documentation for information on managing computer certificates.
- **Error: No signing certificate configured. Please configure a signing certificate**  
**Resolution:** Ensure that a valid certificate and thumbprint have been configured. See [Troubleshooting on page 10](#) for more information.

## DNS or 404 Errors Displayed After Connecting to Adobe Sign

**Issue:** After clicking **Link Account** to connect an Adobe Sign account to OnBase, DNS or 404 errors are displayed.

**Resolution:** Ensure that your **Adobe Sign Hostname URL** field is configured correctly in Studio.

## Cannot Configure Refresh Token

**Issue:** While setting up the refresh token, the following error message is displayed:

**Unable to authorize access because the client configuration is invalid: invalid\_request**

**Resolution:** Ensure that your Adobe API Application redirect URLs are configured correctly. For example:

```
https://machinename/apiserver/esign/esignature/auth/  
callback;https://machinename/apiserver/esign/esignature
```

In addition, you should ensure that the **Client ID** and **Client Secret** for your IdP are configured correctly when connecting your Adobe Sign account to OnBase.

## Cannot Execute Adobe Sign Workflow Rules or Actions

**Issue:** When executing Adobe Sign Workflow rules or actions, the following error messages are displayed:

**Status response from 'https://api.na2.echosign.com/oauth/refresh' was 'Unauthorized'**

**Invalid response from Adobe while attempting to exchange the refresh token. Calling 'https://api.na2.echosign.com/'**

**Resolution:** Ensure that your Adobe API Application redirect URLs are configured correctly. For example:

```
https://machinename/apiserver/esign/esignature/auth/  
callback;https://machinename/apiserver/esign/esignature
```

In addition, you must enable all required scopes for the application. The following scopes are required at minimum:

- **user\_read** (with the **modifier** set to **self**)
- **user\_login** (with the **modifier** set to **self**)
- **agreement\_read** (with the **modifier** set to **group**)
- **agreement\_write** (with the **modifier** set to **group**)

## Cannot Connect to a Data Source

**Issue:** The Integration for Adobe Sign is unable to connect to a data source. In addition, an error message similar to the following may be displayed:

**Invalid or missing required tenant information ElectronicSignatureService:TenantList settings for the database.**

**Resolution:** Ensure that the connection information in the **5\_esignature.json** file has been properly configured (stored in the **ElectronicSignatureService | TenantList** section). This information must match the database connection used by Workflow. You should also check to make sure that all backslash characters are escaped properly to account for JSON file formatting. For example, to configure a data source of **MyDB\SQLInstance**, you would need to enter **Data Source=MyDB\\SQLInstance**.

## Cannot Create Trust Configuration

**Issue:** When clicking the link to create the trust configuration with the HESIS, the following error message is displayed:

**Unable to authorize access because the client configuration is invalid: invalid\_request**

**Resolution:** Ensure that your Adobe's API Application is configured with the correct redirect URIs and scopes. See [Pre-Installation on page 4](#) for more information.

## Cannot Send or Retrieve Adobe Sign Documents

**Issue:** The Integration for Adobe Sign can no longer send or retrieve documents to Adobe Sign. In addition, the following error message is displayed in the Diagnostics Console:

**Invalid response from while trying to exchange refresh token**

**Resolution:** The Adobe refresh token generated when you link Adobe Sign to OnBase is only valid for 6 months. When the token expires, you will need to relink your Adobe Sign account to OnBase to generate a new token. See [Connecting Adobe Sign to OnBase on page 16](#) for information on linking your account.

## Contacting Support

When contacting your solution provider, please provide the following information:

- The OnBase module where the issue was encountered.
- The OnBase version and build.
- The type and version of the connected database, such as Microsoft SQL Server 2014 or Oracle 12c, and any Service Pack that has been installed.
- The operating system that the workstation is running on, such as Windows 10 or Windows Server 2012 R2, and any Service Pack that has been installed. Check the supported operating systems for this module to ensure that the operating system is supported.
- The name and version of any application related to the issue.
- The version of Internet Explorer and any Service Pack that has been installed, if applicable.
- A complete description of the problem, including actions leading up to the issue.
- Screenshots of any error messages.

Supplied with the above information, your solution provider can better assist you in correcting the issue.





# **Integration for Adobe Sign**

## **Administration Guide**

There are three main components to successfully configuring the Integration for Adobe Sign module:

- [Configuring Document Types for Signing on page 15](#)
- [Connecting Adobe Sign to OnBase on page 16](#)

Once you configure these components, you can begin uploading documents to sign using Workflow.

## Pre-Configuration

Before you can use the Integration for Adobe Sign module, you must properly configure your Adobe Sign solution. For complete details on configuring Adobe Sign in general, refer to the help files provided by Adobe Sign.

Additionally, you must have a properly configured OnBase Application Server to successfully connect Adobe Sign with OnBase.

---

**Note:** For more information on setting up the OnBase Application Server, see the **Application Server** module reference guide.

---

## Configuring Document Types for Signing

The Integration for Adobe Sign module allows documents to be uploaded to Adobe Sign for signing. Supported document file formats that can be uploaded to Adobe Sign include PDF, Image Rendered PDF, image, text, and Microsoft Word documents. No specific configuration is required for Document Types to be used with the Integration for Adobe Sign. See the **System Administration** documentation for general information on configuring Document Types.

## Configuring Document Type Auto-Name Strings

Document Auto-Names in OnBase can display up to 255 characters; however, Adobe Sign file names must be 240 characters or less. If Adobe Sign truncates a document Auto-Name because it is more than 240 characters, the document cannot be imported back into OnBase.

Auto-Names for Document Types used for signing should be configured to ensure the document name is always less than 240 characters, regardless of the variables dynamically generated for the Auto-Name when the document is first imported into OnBase.

## Receiving Email Notifications

Successful use of the Integration for Adobe Sign module includes the ability for signers to receive email notifications from Adobe Sign. Ensure users who are to sign documents are able to receive these notifications in their inboxes.

## Connecting Adobe Sign to OnBase

The first step in configuring the Integration for Adobe Sign module is to connect your Adobe Sign account to OnBase. This allows OnBase to establish a link to your Adobe Sign account when uploading and signing documents with Adobe Sign.

---

**Note:** The Adobe refresh token generated when you link Adobe Sign to OnBase is valid for 6 months. When the token expires, you will need to relink your Adobe Sign account to OnBase to generate a new token.

---

To connect Adobe Sign to OnBase, from the OnBase Studio module:

1. From the **Workflow** ribbon, click the **Adobe Sign** button. The **Adobe Sign Integration** dialog box is displayed.

**Adobe Sign Integration**

**Hyland Services**

Electronic Signature Integration Service URL  
[serviceurl]

Hyland IdP URL

Hyland Identity Provider Client ID

**Adobe Sign**

Adobe Sign Hostname URL ⓘ  
secure.[region].echosign.com

Adobe Client ID

**Adobe Sign Account**

No Account Linked

2. Type the URL of your Adobe Sign service in the **Electronic Signature Integration Service URL** field. This field is case-sensitive.

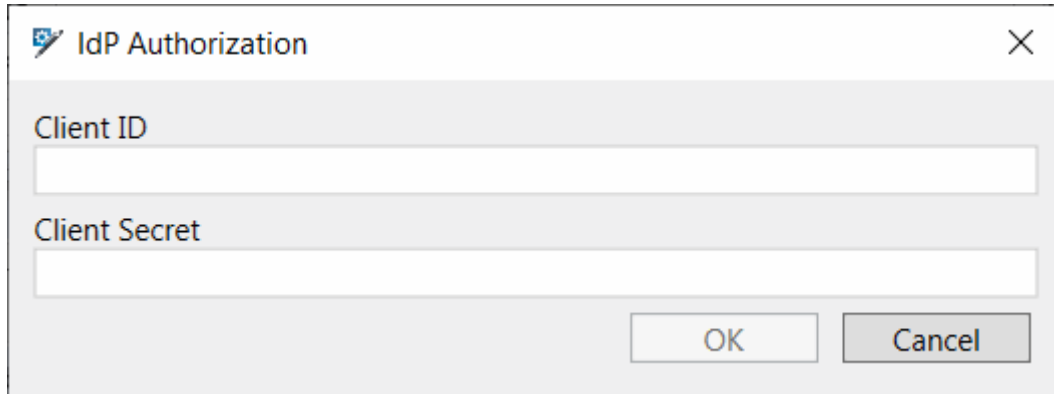
---

**Note:** The URL entered in this field is an API Server endpoint. You must enter a URL that uses the HTTPS protocol in this field.

---

For example: **https://machine.mydomain.com/apiserver/esign**

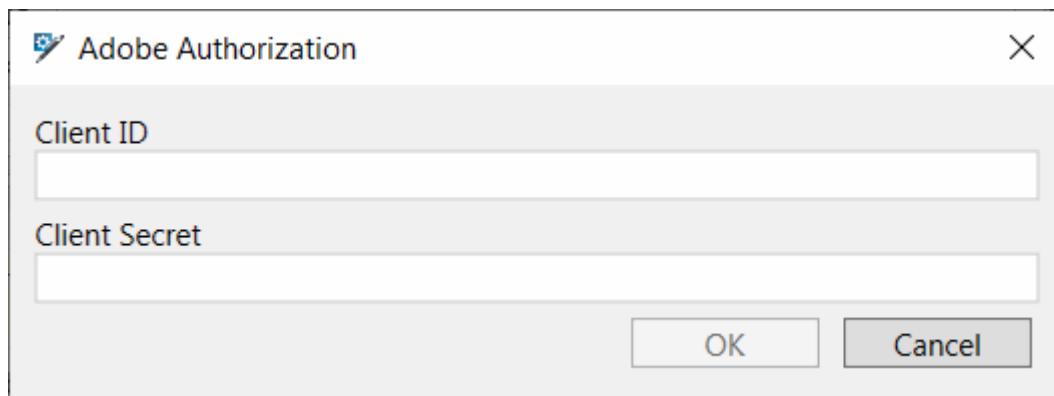
3. Type the URL of your Hyland IdP service in the **Hyland IdP URL** field.
4. Set the client ID of your IdP provider in the **Hyland Identity Provider Client ID** field by clicking **Set...** The **IdP Authorization** dialog box is displayed.

A dialog box titled "IdP Authorization" with a close button (X) in the top right corner. It contains two text input fields: "Client ID" and "Client Secret". At the bottom right, there are two buttons: "OK" and "Cancel".

5. Type the Client ID of the IdP provider in the **Client ID** field.
6. Type the Client Secret of the IdP provider in the **Client Secret** field.
7. Click **OK**.
8. Ensure the correct URL is displayed in the **Adobe Sign Hostname URL** field. This is a required field.

This field must contain your region and is populated using the value configured for your Adobe Sign account. For example: **https://secure.[region].adobesign.com**

See your Adobe Sign documentation for information on configuring this value.
9. Set the client ID of your Adobe Sign configuration in the **Adobe Client ID** field by clicking **Set...** The **Adobe Authorization** dialog box is displayed.

A dialog box titled "Adobe Authorization" with a close button (X) in the top right corner. It contains two text input fields: "Client ID" and "Client Secret". At the bottom right, there are two buttons: "OK" and "Cancel".

10. Type the Client ID of your Adobe Sign configuration in the **Client ID** field.

11. Type the Client Secret of your Adobe Sign configuration in the **Client Secret** field.
12. Click **OK**.
13. Click **Link Account** to link OnBase to the configured Adobe Sign instance. An Adobe Sign login page is displayed.

---

**Note:** Depending on your system's configuration, you may be prompted with a **Security Warning** dialog box. Click **Yes** to allow the Adobe Sign login page to be displayed.

---

14. Enter your Adobe Sign credentials and sign in. Once you have logged in to the Adobe Sign account, it is linked to the Integration for Adobe Sign.
15. Click **OK**.
16. Click **Save** from the **Repository** ribbon group in the **Home** ribbon in order to save your Adobe Sign connection information.

## System Interaction

Certain Workflow rules and actions can be used with the Integration for Adobe Sign module in order to upload and retrieve documents to and from Adobe Sign, as well as to check the status of agreements in Adobe Sign.

## Workflow

When Workflow is licensed, designated rules and actions within Workflow allow you to automate certain processes within the Integration for Adobe Sign module.

- The **Upload to Adobe Sign** action automates uploading of incoming documents to Adobe Sign so that they can be signed. See [Upload to Adobe Sign Action on page 18](#).
- The **Retrieve from Adobe Sign** action retrieves a signed document from Adobe Sign, after it has been uploaded and signed. See [Retrieve from Adobe Sign Action on page 21](#).
- The **Check Agreement Status** rule checks to see if the agreement in Adobe Sign has been signed and completed. See [Check Agreement Status Rule on page 21](#).

## Upload to Adobe Sign Action

Uploads the document and any related documents to the Adobe Sign system. When the action is executed, an Adobe Sign **Compose** page is displayed in your web browser, allowing you to finish and send the agreement within Adobe Sign.

In order to upload documents to Adobe Sign, your Adobe Sign solution must be configured with the **Set a default reminder for agreements created by users in this account** setting enabled. If this setting is not selected, you will not be able to upload documents to Adobe Sign.

---

**Caution:** If you close the browser without clicking **Send** or enabling the **Preview & Add Signature Fields** option and then clicking **Next**, the agreement will not be able to be completed and will be put into the **User Abandoned** status.

---

This action is not supported for use with Unity Scheduler timers.

---

**Note:** If the Auto-Name string for a document being sent is longer than 255 characters, that document's name will be truncated when sent to Adobe Sign.

---



---

**Note:** Any related documents must use supported file types to be able to be uploaded to the Adobe Sign system. Supported file types include .pdf, .doc, .docx, .txt, .rtf, .tif, .jpg, .jpeg, .gif, .png, and .htm. For more information, see your system administrator.

---



---

**Caution:** If the **Upload to Adobe Sign** action is executed on a document multiple times, its existing **Agreement ID** Keyword Value will be overwritten with a new value each time.

---

#### Option: Keyword Type for Adobe Agreement ID

This is a required field.

Select a Keyword Type from this drop-down list. The selected Keyword Type will be used to store the **Agreement ID** Adobe Sign generates after the document(s) to be signed have been sent to Adobe Sign. The Keyword Type to be used must be configured as **Alphanumeric** and have **Mixed Case Values** enabled.

#### Option: Include Related Documents

Select this option to include all related documents when the main document is sent to Adobe Sign. If this option is not selected, only the primary document is sent to Adobe Sign. When this option is enabled, the **Related** tab becomes available so that you can configure which documents are related to others. The options on this tab will vary depending on whether the life cycle containing this action is a Unity life cycle or a non-Unity life cycle.

- For Unity life cycles, configure the options on the **Related** tab as desired:

| Option                           | Description  |
|----------------------------------|--|
| <b>Portfolio Type</b>            | Select this option and select a configured portfolio type from this drop-down list to use all of the relations in the selected portfolio type to find related items.   |
| <b>Portfolio Relation</b>        | Select this option and select a configured portfolio relation from this drop-down list to use the selected portfolio relation to find related items.   |
| <b>Ad Hoc Portfolio Relation</b> | Select this option to configure a new portfolio relation to find related items. Click <b>Configure</b> to open the <b>Portfolio Relation Wizard</b> and configure a relation. See the section on <b>Configuring Portfolio Relations</b> in the <b>Studio</b> documentation for information on using this wizard. |
| <b>Lock Related Item</b>         | Select this option to lock related items along with the primary document.  |

- For non-Unity life cycles, configure the options on the **Related** tab as desired:

| Option   | Description   |
|--|---|
| <b>Document Handle</b>   | Select this option to use the Document Handle to find related documents.  |
| <b>Document Type</b>   | <p>Select this option and select a configured Document Type from this drop-down list to find related documents of the selected Document Type. You must also specify the common Keyword Types to be used to find related documents.</p> <p>To specify common Keyword Types, follow these steps:</p> <ol style="list-style-type: none"> <li>1. Click <b>Common Keywords</b>. The <b>Common Keyword Types</b> dialog box is displayed.</li> <li>2. Select any desired Keyword Types and click <b>Add &gt;&gt;</b>.</li> <li>3. Click <b>OK</b>.</li> </ol> |
| <b>Folder Type</b>   | Select this option and select a configured folder type from this drop-down list to be used to find related items.   |
| <b>Associated Primary Document</b>                             | <p>Select this option to use the selected inbox item as the related item.</p> <p>This option is only valid when executing on a document in the work folder.</p>   |
| <b>Require All Common Keywords Present on Primary Document</b> | <p>Select this option to require that all configured Common Keywords are present on the primary document in order to search for related documents.</p> <p>This option is only available if the <b>Document Type</b> or <b>Folder Type</b> option is selected.</p>   |

#### Option: Get Agreement Name from Property

You can customize the way email signer notifications are displayed to the user using Workflow properties. To set the agreement name from a property value, enter the appropriate property name in the **Get Agreement Name from Property** field. If this option is not selected, the document's Auto-Name string will be used as the Agreement Name.

#### Option: Get Agreement Message from Property

You can customize the way email signer notifications are displayed to the user using Workflow properties. To set the agreement message body from a property value, enter the appropriate property name in the **Get Agreement Message from Property** field. If this option is not selected, the following default value is used: **Please review and complete this document.**

**Option: Auto-Populate Recipients from Property**

You can customize the way email signer notifications are displayed to the user using Workflow properties. To set the email addresses of the notification message from a property value, enter the appropriate property name in the **Auto-Populate Recipients from Property** field. If this option is not selected, the user will be required to fill in the appropriate email addresses when the agreement is created.

---

**Note:** If invalid email addresses are supplied, an error will be displayed when the task is executed and the user will be required to manually correct/enter the recipients email addresses.

---

---

**Tip:** Properties are defined using the **Set Property Value** and **Set Multiple Property Values** actions.

---

## Retrieve from Adobe Sign Action

Retrieves a signed document from Adobe Sign, after it has been uploaded and signed. The retrieved document will be archived in OnBase as a new document of the specified Document Type.

**Option: Document Type**

This is a required field.

Select a Document Type from this drop-down list. The selected Document Type will be used to store documents retrieved from Adobe Sign.

**Option: Keyword Type**

This is a required field.

Select a Keyword Type from this drop-down list. The selected Keyword Type will be used to store the **Agreement ID** Adobe Sign generates after the document(s) to be signed have been sent to Adobe Sign.

## Check Agreement Status Rule

Check to see if the agreement has been returned from the Adobe Sign system as either completed (signed) or declined. Documents are only brought back from Adobe Sign in the event that all documents uploaded as part of a single agreement are completed and/or declined.



When this rule evaluates true, the agreement has been returned as well.

**Tip:** It is not recommended to use the **Check Agreement Status** rule immediately after the **Upload to Adobe Sign** action is executed. The **Upload to Adobe Sign** action requires the user to use the Adobe Sign interface to finish and send the Agreement, so checking the agreement status before this user interaction is completed will return the **User Abandoned** or **Draft** status.

#### Option: Keyword Type

This is a required field.

Select a Keyword Type from this drop-down list. The selected Keyword Type will be used to store the **Agreement ID** of the agreement being evaluated.

#### Option: Agreement Status

Select an agreement status value from this drop-down list. Each agreement status corresponds to one or more agreement status values within Adobe Sign. Agreements whose Adobe Sign status corresponds to the selected value will be evaluated as **True** by the rule and can then have some action performed upon them.

Available status values are:

| Agreement Status          | Description  |
|---------------------------|--|
| <b>Completed</b>          | Applied to agreements with an Adobe Sign status of <b>ACCEPTED</b> , <b>APPROVED</b> , or <b>SIGNED</b> . This option is selected by default.  |
| <b>Awaiting Signature</b> | Applied to agreements with an Adobe Sign status of <b>OUT_FOR_SIGNATURE</b> , <b>OUT_FOR_ACCEPTANCE</b> , <b>OUT_FOR_APPROVAL</b> , or <b>OUT_FOR_DELIVERY</b> .                                     |
| <b>Cancelled</b>          | Applied to agreements with an Adobe Sign status of <b>CANCELLED</b> .  |
| <b>Expired</b>            | Applied to agreements with an Adobe Sign status of <b>EXPIRED</b> .  |
| <b>Draft</b>              | Applied to agreements with an Adobe Sign status of <b>DRAFT</b> .  |
| <b>User Abandoned</b>     | Applied to agreements that have been abandoned by the user within Adobe Sign (for example, if the user closes the Adobe Sign <b>Compose</b> page without sending the agreement or saving the draft). |



# Integration for Adobe Sign

## User Guide

The Integration for Adobe Sign module allows you to upload documents from OnBase to the Adobe Sign service. Once an agreement is uploaded to Adobe Sign, Adobe Sign generates email notifications and sends them to the required signers. Once a document is signed by all required parties, the document can be returned to OnBase for long-term storage.

Upload and retrieval is performed automatically using OnBase Workflow. No manual user interaction is required.