File Conversion Service

Installation Guide

Version: 8.x.x

Written by: Documentation Team, R&D

Date: Friday, October 10, 2025

Documentation Notice

The information and software described in this document are furnished only under a separate agreement and may only be used or copied according to the terms of such agreement. It is against the law to copy the software except as specifically allowed in such agreement. 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. and/ or one of its affiliates.

Hyland, OnBase, Alfresco, Nuxeo, Content Innovation Cloud and other product or brand 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

The information in this document may contain technology as defined by the Export Administration Regulations (EAR) and could be subject to the Export Control Laws of the U.S. Government including for the EAR and trade and economic sanctions maintained by the Office of Foreign Assets Control as well as the export controls laws of your entity’s local jurisdiction. Transfer of such technology by any means to a foreign person, whether in the United States or abroad, could require export licensing or other approval from the U.S. Government and the export authority of your entity’s jurisdiction. You are responsible for ensuring that you have any required approvals prior to export.

DISCLAIMER: This documentation contains available instructions for a specific Hyland product or module. This documentation is not specific to a particular customer or industry. All data, names, and formats used in this document’s examples are fictitious unless noted otherwise. This document may reference websites operated by third parties. In such a case, Hyland has no control or liability for the content of such third-party websites. The inclusion of such a link shall not constitute an endorsement or affiliation with such a third-party website; the reference is provided for information purposes only. If you have questions about discrepancies in this document, please contact Hyland. Hyland customers are responsible for making their own independent assessment of the information in this documentation. This documentation: (a) is for informational purposes only, (b) is subject to change without notice, (c) is confidential information of Hyland Software, Inc. and its affiliates and (d) does not create any commitments or assurances by Hyland. This documentation is provided “as is” without representation or warranty of any kind. Hyland expressly disclaims all implied, express, or statutory warranties. Hyland’s responsibilities and liabilities to its customers are controlled by the applicable Hyland agreement. This documentation does not modify any agreement between Hyland and its customers.

Table of Contents

What is File Conversion Service?

Perceptive Content File Conversion Service (FCS) is a Web API used with Integration Server that allows for the rendering and display of raster images from Perceptive Content or third-party sources.

Through the GET: /document/{id}/page/{pageID}/rendition Integration Server call, you can request the rendering of raster content from Perceptive Content. The call retrieves a PNG rendition of a document page, and the request contains the document ID and page ID for the page that you want to convert.

Install and configure

Download and run File Conversion Service in a Windows environment

During the installation of the File Conversion Service, you must perform the following procedures.

Prerequisite You must have Node.js 64-bit installed. Refer to the File Conversion Service Technical Specifications Guide for the appropriate version. For downloads and instructions on installing, refer to the Node.js website.

When installing the service make sure to conduct a clean installation rather than overwriting previously installed files. Delete any previous versions of Perceptive Content File Conversion Service. A clean installation is required for the service to run correctly.

1. Download the Perceptive Content File Conversion Service file for the Windows 64-bit platform to a local directory.
2. Unzip the files from the download location.
3. To install as a service, navigate to the *file-image-service* directory and type `node file-image-service install`.
4. To start the service, type `node file-image-service start`.
5. Optional. Configure the service by editing the *config.json* file.

   You can configure conversion options such as DPI, TIFF compression, and JPEG quality in the *ISYS11df.ini*file.

Install File Conversion Service on Red Hat Enterprise Linux 7 and Oracle Linux 7

To install File Conversion Service on a system running Red Hat Enterprise Linux (RHEL) 7 or Oracle Linux 7, complete the following steps.

Prerequisite You must have Node.js 64-bit installed. Refer to the File Conversion Service Technical Specifications Guide for the appropriate version. For downloads and instructions on installing, refer to the Node.js website.

When installing the service, make sure to conduct a clean installation rather than overwriting previously installed files. Delete any previous versions of File Conversion Service. A clean installation is required for the service to run correctly.

1. Download *file-image-service-6.x.x-linuxtarball.tar.gz* to a local directory.
2. Unzip the files from the download location.
3. To install as a service, type `node file-image-service install`.

4. To start the service, type `node file-image-service start`.
5. Optional for RHEL 7 and Oracle Linux 7. To configure the service to start when the system boots, type `systemctl enable perceptivecontentfileconversionservice.service`.
6. Optional. Configure conversion options such as DPI, TIFF compression, and JPEG quality in the *ISYS11df.ini* file.

Upgrade File Conversion Service in a Windows environment

To upgrade File Conversion Service in a Windows environment, complete the following steps.

1. Navigate to the *file-image-service* directory.
2. To stop the currently installed service, type `node file-image-service stop` from the *file-image-service* directory.
3. Back up the config.json file in the *file-image-service* directory for reference after the installation.
4. To uninstall the currently installed service, type `node file-image-service uninstall` from the *file-image-service* directory.
5. Delete the currently installed version of file-image-service.
6. Download and install the version of Node.js specified in the File Conversion Service Technical Specifications Guide. For downloads and instructions on installation, refer to the Node.js website.
7. Download the Perceptive Content File Conversion Service file for the Windows 64-bit platform to a local directory.
8. Unzip the files from the download location.
9. To install as a service, navigate to the file-image-service directory and type `node file-image-service install`.
10. To start the service, type `node file-image-service start`.
11. Optional. Configure the service by editing the config.json file. You can configure conversion options such as DPI, TIFF compression, and JPEG quality in the ISYS11df.ini file.

Upgrade File Conversion Service in a Linux environment

To upgrade File Conversion Service in a Red Hat Enterprise Linux (RHEL) or Oracle Linux 7 environment, complete the following steps.

1. Navigate to the *file-image-service* directory.
2. To stop the currently installed service, type `node file-image-service stop` from the *file-image-service* directory.
3. Back up the config.json file in the *file-image-service* directory for reference after the installation.
4. To uninstall the currently installed service, type `node file-image-service uninstall` from the *file-image-service* directory.
5. Delete the currently installed version of file-image-service.
6. Download and install the version of Node.js specified in the File Conversion Service Technical Specifications Guide. For downloads and instructions on installation, refer to the Node.js website.
7. Download *file-image-service-6.x.x-linuxtarball.tar.gz* to a local directory.
8. Unzip the files from the download location.

9. To install as a service, navigate to the file-image-service directory and type `node file-image-service install`.
10. To start the service, type `node file-image-service start`.
11. Optional. To configure the service to start when the system boots, type `systemctl`enable perceptivecontentfileconversionservice.service.
12. Optional. Configure the service by editing the config.json file. You can configure conversion options such as DPI, TIFF compression, and JPEG quality in the ISYS11df.ini file.

config.json properties

The following list provides information about the properties defined in the File Conversion Service *config.json* file. The *config.json* file contains parameters that can be defined by an administrator.

The properties listed are in the order they appear in the *config.json* file.

port

Specifies the network port on which the service accepts the request. The default is 1337.

loggingLevel

Specifies the level of logging messages to save. The default is warn.

Entry examples include the following values.

- verbose
- info
- warn
- error

logLocation

Specifies the path for the log file storage. The default is ./logs

Note: Backslash characters must be escaped with a preceding “ \”. For example, *C:\\file-image-service\\logs\\*

loadAverageLoggingInterval

Specifies an interval in seconds to log load averages for conversion requests to the console. Load averages will not be logged if the interval is less than or equal to 0. The default is 0.

uploadLocation

Specifies the path for uploaded file storage. The default is ./uploads.

Note: Backslash characters must be escaped with a preceding “ \”. For example, *C:\\file-imageservice\\uploads*

convertedFileLocation

Specifies the path for converted file storage. The default is ./converted.

Note: Backslash characters must be escaped with a preceding " \". For example, *C:\\file-imageservice\\converted\\*

conversionLifetimeInMinutes

Specifies the number of minutes that a converted file is stored. The minimum value is 1. The default is 10.

maxConcurrentConversions

Specifies the number of concurrent requests that can be handled by the File Conversion Service instance. This also throttles the number of async jobs that processed simultaneously.

accessLogFormat

Specifies the access log output format. Defaults to the following:

```
:remote-addr - :remote-user [:date[clf]] ":method :url HTTP/:http-
version" :status :res[content-length] ":referrer" ":user-agent" :req
[content-length] - :response-time :total-time
```

trustReverseProxy

Specifies whether the client's IP address, request hostname, and request protocol should be derived from X-Forwarded-* headers set by a trusted reverse proxy. If disabled, the request will be used directly to determine the client IP, hostname and protocol. When enabled it is important to ensure a trusted reverse proxy is removing or overwriting the `X-Forwarded-For`, `X-Forwarded-Host` and `X-Forwarded-Proto` headers, otherwise a client may specify any value. The default is true.

statefulApiEnable

Specifies whether stateful APIs should be enabled. These APIs will be subject to removal in a future release. Consumers should be updated to use stateless conversion APIs. The default is false.

connectionKeepAliveTimeoutSeconds

Specifies the number of seconds of inactivity the server needs to wait for additional incoming data before the socket will be closed. A value of 0 will disable the keep-alive timeout behavior. The default is 30.

secureHttpCert

Specifies an X.509 certificate chain in PEM format, used to configure TLS. Must include a certificate for a provided PEM private key.

secureHttpCertFilePath

File path to an X.509 certificate chain in PEM format, used to configure TLS. Must include a certificate for a provided PEM private key.

secureHttpKey

Specifies a private key in PEM format, used to configure TLS.

secureHttpKeyFilePath

File path to a private key in PEM format, used to configure TLS.

secureHttpPfxEncoded

Specifies a Base-64 encoded PFX or PKCS12 private key and certificate chain, used to configure TLS.

secureHttpPfxFilePath

File path to a PFX or PKCS12 encoded private key and certificate chain, used to configure TLS.

secureHttpPassphrase

The passphrase used for encrypted private keys or PFX keystores.

If this option is used, `secureHttpPassphraseFilePath` must not be specified.

secureHttpPassphraseFilePath

File path to a file containing the passphrase used for encrypted private keys or PFX keystores.

If this option is used, `secureHttpPassphrase` must not be specified.

secureHttpMinVersion

The minimum TLS version to allow. Avoid setting this lower than TLSv1.2. Allowed values are `TLSv1.3`, `TLSv1.2`, `TLS1.1` or `TLSv1`. The default is TLSv1.2.

secureHttpMaxVersion

The maximum TLS version to allow. Allowed values are `TLSv1.3`, `TLSv1.2`, `TLS1.1` or `TLSv1`. The default is the NodeJS default TLS maximum version, which is currently TLSv1.3.

secureHttpSessionTimeout

The number of seconds after which a TLS session can no longer be resumed. The default is 300.

File Conversion service commands

Use the following commands to manage File Conversion Service as a Windows service or Linux daemon. These commands are supported for Windows, Red Hat Enterprise Linux version 7 and Oracle Linux version 7.

Navigate to the *file-image-service* directory you created during installation to run commands.

Syntax

```
node file-image-service [options] [command]
```

start

Use the `start` command to start the service.

```
node file-image-service start
```

stop

Use the `stop` command to stop the service.

```
node file-image-service stop
```

install

Use the `install` command to install the service

```
node file-image-service install
```

uninstall

Use the `uninstall` command to uninstall the service.

```
node file-image-service uninstall
```

Options

`--help`

Use the `--help` option to display usage information.

`--version`

Use the `--version` option to display the version number.

Appendix A: Perceptive Content Integration Server and File Conversion Service deployment

File Conversion Service deployment

You should base File Conversion Service deployment on anticipated rendering loads across associated instances of Integration Server. You may encounter rendering issues if the number of concurrent rendering requests sent to File Conversion Service by Integration Server exceeds the configurable maximum number of conversions in the File Conversion Service service. By default, the maximum number of conversions in File Conversion Service is based on the number of logical processors available on the system.

By default, Integration Server also sends a maximum of eight concurrent rendering requests to File Conversion Service . Each document page rendering operation uses a CPU core on the File Conversion Service server. If the number of concurrent rendering requests is greater than the number of available CPU cores, the system returns a 500 server error for that operation. When this occurs, documents and pages fail to render and the system displays the message, *"The preview is not available"*.

Configure concurrent rendering operations

Each File Conversion Service node limits the number of concurrent operations it can handle at one time. The **maxConcurrentConversions** property controls this behavior. If this number of conversions is exceeded at the time a request is made the request fails. If users are consistently exceeding the **maxConcurrentConversions** available, updates should be made to File Conversion Service to ensure that it can handle the request load. You can increase the **maxConcurrentConversions** on a single node based on the CPU utilization for the File Conversion Service service and overall availability of system resources. Though vertical scaling of File Conversion Service is possible, most environments require pooling File Conversion Service resources across multiple hosts. We recommend putting File Conversion Service behind a load balancer that supports **least concurrent requests** balancing algorithms. Round robin load balancing does not guarantee that your File Conversion Service pool is used to its full potential. For more information on this process, refer to the File Conversion Service Online Help.

Update Integration Server connection pool settings

By default, Perceptive Content Integration Server uses a connection pool for FCS requests. This ensures that connections do not have to be re-established for every request, but it also limits the number of concurrent requests that can be made to FCS. If you adjust the number of conversions that the FCS pool can do you must also adjust the render.fileimageservice.maxrequests parameter to make use of those resources. If you have multiple instances of Perceptive Content Integration Server, each instance should be configured to use a subset of the available resources so the FCS pool is not oversubscribed. If you have a single instance of Integration Server, then this number should match the cumulative number of maxConcurrentConversions available in your FCS pool.

You can use the following expression as starting point.

```
Integration Server FCS Max Requests per Instance = FCS
maxConcurrentConversions x Number of FCS instances / Number of Integration
Server Instances
```

To configure the Integration Server FCS Max Requests settings complete the following steps.

1. On the Integration Server web server, go to the *integrationserver\WEB-INF* directory, and then open the **integrationserver.ini** configuration file in a text editor
2. Locate the **render.fileimageservice.maxrequests** parameter, and adjust it based on the number of concurrent requests the FCS pool supports.
3. Save and then close the **integrationserver.ini**configuration file.
4. Restart application server for the change to take effect.

File Conversion Service endpoints

Integration Server relies on a File Conversion Service pool to perform the following operations.

Operation	Setting	Description
Page rendition	/document/{id}/page/{pageId}/rendition	Use this operation to view pages from Perceptive Experience clients.
Document export	/document/export	Use this operation to export documents from Perceptive Content.
Document conversion	/document/{id}/conversion/{outputType}	Use this operation to print documents from Perceptive Experience clients.

If you improperly configure an File Conversion Service pool, the Perceptive Experience client can display the message, “The preview is not available”, on random document pages. This message may also display on printouts from Experience directly, on print previews, or when printing from an associated application such as Adobe Acrobat or Foxit.

Additional considerations

You can create multiple tiers of Integration Server and File Conversion Service pools to ensure that Experience applications are not starved by automated processes. This requires updating the environment to have different application server tiers and isolated File Conversion Service pools for each group of Integration Server resources. You can direct Experience clients to the application tier and the automated processes to the API tier through routing.

Appendix B: TLS configuration

File Conversion Service can be configured to use Hypertext Transfer Protocol Secure (HTTPS) and supports multiple TLS versions. A private key and certificate are required to configure TLS. A private key and certificate can be specified using discrete PEM encoded files, or as a single PFX or PKCS12 keystore. Encrypted private keys in either PEM or PKCS12 form can be used by providing a passphrase either via a configuration setting, environment variable or file containing the passphrase.

Examples

The following is an example of TLS configuration using PEM private key and certificate files listening on port 8443

```
{

    "port": "8443",
    "loggingLevel": "warn",
    "logLocation": "./logs",
    "uploadLocation": "./uploads",
    "convertedFileLocation": "./converted",
    "connectionKeepAliveTimeoutSeconds": 30,
    "conversionLifetimeInMinutes": "10",
    "trustReverseProxy": true,
    "statefulApiEnable": false,
    "secureHttpCertFilePath": "./tls.crt",
    "secureHttpKeyFilePath": "./tls.key",
    "secureHttpPassphrase": "tlspassword"
}
```

The following is an example of TLS configuration using a PFX or PKCS12 keystore listening on port 8443.

```
    {
        "port": "8443",
        "loggingLevel": "warn",
        "logLocation": "./logs",
        "uploadLocation": "./uploads",
        "convertedFileLocation": "./converted",
        "connectionKeepAliveTimeoutSeconds": 30,
        "conversionLifetimeInMinutes": "10",
        "loadAverageIntervalSeconds": "60",
        "trustReverseProxy": true,
        "statefulApiEnable": false,
        "secureHttpPfxFilePath": "./tls.p12",
        "secureHttpPassphraseFilePath": "./tlspassphrase.secret"
    }
```

The following is an example of TLS configuration with minimum and maximum TLS versions listening on port 8443.

```
{
    "port": "8443",
    "loggingLevel": "warn",
    "logLocation": "./logs",
    "uploadLocation": "./uploads",
    "convertedFileLocation": "./converted",
    "connectionKeepAliveTimeoutSeconds": 30,
    "conversionLifetimeInMinutes": "10",
    "loadAverageIntervalSeconds": "60",
    "trustReverseProxy": true,
    "statefulApiEnable": false,
    "secureHttpPfxFilePath": "./tls.p12",
    "secureHttpPassphraseFilePath": "./tlspassphrase.secret",
    "secureHttpMinVersion": "TLSv1.2",
    "secureHttpMaxVersion": "TLSv1.3"
}
```