Perceptive Integration Server on Tomcat

Installation and Setup Guide

Version: 7.1.x

Written by: Product Knowledge, R&D
Date: March 2020

Hyland

Copyright

Table of Contents

Overview

Perceptive Integration Server is a service that provides a window into Perceptive Content functionality. Integration Server exports a rich set of web services that enable customers and third-party application developers to embed Perceptive Content functionality, such as document management and workflow, directly into their applications.

For product technical specifications and system requirements, see the *Product Technical Specifications* for your product version.

Prepare to install

To install Integration Server, use the platform-specific steps for your operating system. The “Install Integration Server” section of this document provides all installation methods.

System Requirements

Integration Server follows the same minimum system requirements as Perceptive Content Server. For product technical specifications and system requirements, see the *Technical Specifications* documentation available on the Customer Portal at www.perceptivesoftware.com.

Note Before you install Integration Server, you must have Apache Tomcat successfully installed as the web application server.

Compatibility

Integration Server 7.1 is not backwards compatible with earlier versions of ImageNow. Your Integration Server version must be the same version as Perceptive Content Server.

Licensing requirements

Integration Server is licensed on a transaction basis. The use of the agent requires a license that provides a certain number of transactions over a period.

If you are installing Integration Server for use with an ImageNow Interact product that is not licensed by transactions, you only need the license for that specific Interact product. If the Interact product is licensed by transactions, you must have an Integration Server license as well as the specific Interact license. For more licensing information, see the product’s installation and setup guide.

The following table contains three example scenarios for Integration Server licensing.

Scenario	Required Licenses	Number of licenses	Description
User developed application	Integration Server license Integration Server transaction pack license	A minimum of one Integration Server license. A minimum of one Integration Server transaction pack license.	The Integration Server transaction pack license must contain enough transactions to cover the number of calls you expect your users to make collectively per transaction period and 300 transactions per hour. For example, if 30 users make 10 calls per hour each, you need at least two transaction packs per hour specified on the license.

Scenario	Required Licenses	Number of licenses	Description
Using an Integration Server compatible product that uses a seat or concurrent license.	Specific license for that product only.	See the product's documentation.	You do not need the Integration Server license or Integration Server transaction pack license, because all of the calls are covered by the product's specific license. **Note** Integration Server must be installed.
Using an Integration Server compatible Interact product that uses a transaction pack scheme.	Integration Server license Transaction pack license for the product	One Integration Server license. A minimum of one license for the product.	The license for the product must have enough transaction packs for the number of transactions your users need to make. **Note** For this scenario, you do not need an Integration Server transaction pack license.

Integration Server installation

This document assumes you are installing Perceptive Integration Server for the first time. The steps for installing are listed below.

- Download the Integration Server files
- Deploy Integration Server
- Configure Integration Server

Download the Integration Server files

Starting with 7.0, Integration Server is provided as a WAR file. This allows easy deployment to web application servers, such as Apache Tomcat. To download the Integration Server files, complete the following steps.

1. Go to www.perceptivesoftware.com and log in to the Customer Portal.
2. In the **Product Downloads** page, search for all downloadable items for the specific product and version you want to use. These files may include a product installer, product documentation, or set of supporting files.
3. Download the relevant files to a temporary directory on your computer.

Deploy Integration Server

Use this procedure to deploy the downloaded Integration Server web application into your environment.

1. Stop Apache Tomcat
2. Move the **integrationserver.war** file to **${TOMCAT_HOME}/webapps** directory.
3. Start Apache Tomcat, which will automatically unzip and deploy the integrationserver.war file.

   Note You can use other procedures to deploy a WAR file to Apache Tomcat.

Configure Integration Server

Within integrationserver.ini file settings, you can configure such things as the IP address, port, and logging for Integration Server. You can use the settings in the integrationserver.ini file to customize how Integration Server works.

Note When you make changes to Integration Server's INI file, restart Tomcat for the new changes to be picked up.

In most cases, you need to modify only the IP address and port for the Perceptive Content Server. Integration Server communicates with the Perceptive Content Server using these settings.

1. In the **[path]/${TOMCAT_HOME}/webapps/integrationserver/WEB-INF** directory, select the **integrationserver.ini** file and then open it in a text editor.
2. In the **[General]** section, change the settings as needed using the following substeps.
    1. To change the value for the server address, provide an IP address for the **server.ip.address** setting. The default is **127.0.0.1**.
    2. To change the server port, enter a value for the port for the **server.port** setting. The default is **6000**.
3. In the **[Logging]** section, set the logging for the following settings. A detailed explanation of each setting is provided in the **integrationserver.ini** setting table in Appendix A.
    - **log.directory**. Specifies the path to the directory that contains the log files. The default is **logs**.
    - **log.rolling.interval**. This setting specifies the amount of time data is stored in a log file before the data is archived to another file. This keeps log files from becoming too large. The default is **DAY**.
    - **log.max.history**. This setting is a threshold for the maximum number of log files that are retained per day. One file is created each day, hour, or month depending on the log.rolling.interval setting. The default is **28**.
4. In the [**Capture]** section, set the directory to use for document captures and transfers. For example, **capture.work.directory=[*drive*:]/inserver/temp**. The default is **temp**.
5. Save the file and then close it.

Configure Integration Server for single sign-on

Single sign-on (SSO) allows users to access Integration Server by authenticating with an external identity provider. Currently, Perceptive Experience is the only Perceptive product that supports Integration Server's SSO configuration.

Note When single sign-on is enabled, your single sign-on provider supplies authentication information. Users cannot log in to Integration Server anonymously.

Generate a keystore and key pair

To establish trust between Integration Server and Perceptive Content Server, you must generate a keystore and a key pair. To generate a keystore and a key pair, complete the following steps.

Note Perform these steps on your Integration Server machine.

1. Open a command window and navigate to the bin directory of your JRE installation.
2. Enter the following command and press ENTER.

```
keytool -genkeypair -alias [alias] -keyalg [RSA] -keystore [path to keystore] -keysize
[key size]
```

Setting	Options	Description
-alias	Any valid alias.	Specifies the identifier associated with the key pair to be generated in the output key store.
-keyalg	RSA	Specifies the algorithm to be used to generate the key pair.
-keystore	Any valid keystore path.	Specifies the keystore path. **Note** Only JKS keystores are supported.
-keysize	Any valid key size.	Specifies the size of each key to be generated. For example, 512, 1024, 2048.

Example

```
keytool -genkeypair -alias PSWProd1 -keyalg RSA -keystore C:\Keystores\PSWProd1.jks -
keysize 1024
```

Export a certificate

After generating a keystore and a key pair, a certificate is created. To export your certificate to be imported into Perceptive Content Server, complete the following steps.

1. Open a **Command Prompt** window and navigate to the bin directory of your JRE installation.
2. Type the following command.

```
keytool –exportcert –keystore [path to keystore] –alias [alias] –file [certificate
file to create]
```

Example

```
keytool -exportcert -keystore C:\Keystores\PSWProd1.jks -alias PSWProd1 -file
C:\Keystores\PSWProd1.cer
```

Import a certificate into Perceptive Content Server

To import a certificate into Perceptive Content Server, complete the following steps.

1. Copy the certificate and paste it into the **etc** directory on the Perceptive Content Server machine.
2. Open a **Command Prompt** window and navigate to the **bin/bin64** directory.
3. Type the following command.

```
intool --cmd import-cert --file [drive]:\{certificate location} --type pki-sso
```

Configure Integration Server to use a keystore

To configure Integration Server to use a keystore, complete the following steps.

1. In the **WEB-INF** folder of your Integration Server deployment, open the **integrationserver.ini** file in a text editor.
2. Set **sso.enabled** to **TRUE**.
3. Set **sso.keystore.location** to **C:/Keystores/PSWProd1.jks**.
4. **Optional** Enter the keystore password using the **sso.keystore.password** setting to verify the integrity of the keystore.
5. Enter the key pair alias using the **sso.privatekey.alias** setting.
6. Enter the key password using the **sso.privatekey.password** setting.
7. Set the **sso.authenticator.class** to the fully qualified class used to determine authentication from an external provider.

   Note You can use the built-in authentication class **com.imagenow.authentication.translator.HTTPHeaderTranslator** or create your own authentication class.
8. Configure the authentication class settings.

   Note If using the built-in authentication class, set the **sso.httpheader.name** value to the authorized user header.

SSL/TLS considerations

If you want to configure SSL/TLS to use with Integration Server, you must do so on your Tomcat web application server. We recommend using a CA signed certificate. For more information, see the Tomcat help documentation on the Apache website.

To establish trust between Integration Server and Perceptive Content Email Agent, when Integration Server is set up for SSL/TLS, you must import the Integration Server certificate into the Email Agent trustore. To establish trust with Email Agent, see the Email Agent Installation Guide.

Appendix: A integrationserver.ini file options

The following table provides definitions and sample data for the settings in the integrationserver.ini configuration file. This table displays the INI settings under group headings in brackets, for example, [General], in the order the groups appear in the INI file. Each setting offers two or more options, which are defined in the table below along with a description of each setting and its options. Use this table as a guide when customizing the file.

Group	Setting	Options	Description
General	server.ip.address	Any valid and explicit IP address	The IP address of the Perceptive Content server to use. For example, server.ip.address=localhost
	server.port	Any valid port.	Specifies the port number of the Perceptive Content Server. For example, server.port=6000 The default is 6000.
	session.cookie.samesite.policy	UNSET NONE LAX STRICT	Specifies the SameSite policy set on session cookies. The default is UNSET.
	session.timeout	Any positive number	Specifies the connection timeout in minutes. For example, session.timeout=60 The default is 60.
	connection.timeout.logon	Any positive number	Specifies how many seconds to wait for successful login before terminating the connection. The default is 30.
	connection.timeout.search	Any positive number	Specifies how many seconds to wait for search calls before terminating the connection. The default is 30.
	connection.timeout.erm	Any positive number	Specifies how many seconds to wait for ERM before terminating the connection. The default is 30.
	connection.timeout.default	Any positive number	Specifies how many seconds to wait for server calls before terminating the connection. The default is 30.
SSO	sso.enabled	TRUE FALSE	This setting specifies whether Single Sign On is enabled. The default is FALSE.

Group	Setting	Options	Description
	sso.keystore.location	Any valid file path location.	This setting specifies the file path location of the key store to be used for Single Sign On authentication. A forward slash should be used as the file separator.
	sso.keystore.password	At least 6 characters.	This optional setting specifies the password to verify the integrity of the key store.
	sso.privatekey.alias	Any valid string.	This setting specifies the alias for the key pair to be used for Single Sign On authentication.
	sso.privatekey.password	At least 6 characters.	This setting specifies the password for the private key to be used for Single Sign On authentication.
	sso.authenticator.class	Any valid fully qualified class.	This setting specifies the fully qualified class used to determine authorization from an external provider. This class must be in the application class path. Optional built-in classes provided for this purpose are: **com.imagenow.authentication.translator.HTTPHeaderTranslator** **com.imagenow.authentication.translator.x509ldap.X509LDAPAuthenticationTranslator**
	sso.httpheader.name	Any valid HTTP header field name.	This setting specifies the name of the HTTP request header to be used by the built-in class HTTPHeaderTranslator plugin to map authentication information from an external provider.
Logging	log.level	0 through 7	This setting specifies the verboseness of the logging that Integration Server uses to log errors for troubleshooting. The higher the number, the more verbose the logging. Typically, set a minimal logging level unless debugging an issue. If logging is increased, make sure the logging level is reset to a lower level after the debugging is finished. Failure to do so can greatly affect performance and disk space. 0 = Logging is turned off (do not use) 1 = FATAL 2 = ERROR 3 = WARN 4 = INFO 5 = DEBUG 6 = TRACE 7 = ALL A logging level logs all information for that level and the lower levels. For example, if logging is set to 4, logging is turned on for levels 4, 3, 2, and 1. The default is 2.

Group	Setting	Options	Description
	log.level.thirdparties	0 through 7	Specifies the level Integration Server uses to logs messages from third-party libraries, such as the Spring Framework. The higher the number, the more verbose the logging. Typically, set a minimal logging level unless debugging an issue. If logging is increased, make sure the logging level is reset to a lower level after the debugging is finished. Failure to do so can greatly affect performance and disk space. 0 = Logging is turned off (do not use) 1 = FATAL 2 = ERROR 3 = WARN 4 = INFO 5 = DEBUG 6 = TRACE 7 = ALL A logging level logs all information for that level and the lower levels. For example, if logging is set to 4, logging is turned on for levels 4, 3, 2, and 1. The default is 2.
	log.timing.enabled	TRUE FALSE	Specifies if timing logging should be enabled. Timing logging outputs a small message when a request or response is transmitted. FALSE = Timing logging is disabled TRUE = Timing logging is enabled The default is FALSE.
	log.interceptor.enabled	TRUE FALSE	Specifies if interceptor logging should be enabled. Interceptor logging outputs the full request and response messages transmitted, which aids in debugging. FALSE = Interceptor logging is disabled TRUE = Interceptor logging is enabled The default is FALSE.

Group	Setting	Options	Description
	log.line.pattern	Any conversion pattern (including literal text) using valid specifiers and modifiers.	This setting formats the output of a logging event as a string of literal text. Each conversion specifier starts with a percent sign (%) and is followed by optional format modifiers and a conversion pattern name. The conversion pattern name specifies the type of data, such as logger, level, date, and so on. The format modifiers control such things as field width, padding, and justification. **Note** This pattern adheres to the Log4j PatternLayout standard.
	log.directory	Any valid path	Specifies the path to the directory that contains the log files. It is assumed that the Integration Server is installed and running on the same machine as Perceptive Content Server. The default is C:/inserver/logs
	log.rolling.interval	MONTH, DAY, or HOUR	This setting specifies the amount of time data is stored in a log file before the data is archived to another file. This keeps log files from becoming too large. The default is DAY.
	log.max.history	Any positive number.	This setting is a threshold for the maximum number of log files that are retained. One file is created each day, hour, or month depending on the log.rolling.interval setting. Note This setting excludes the log file that is currently being used to record events. For example, if log.max.history is set to 3 and the log.rolling.interval is set to DAY, then today's log file plus three additional log files exist for four log files. The default is 28. Giving this setting a value of 0 indicates that there should be no maximum number of log files retained. In other words, log files will not be automatically deleted.
Capture Settings	capture.work.directory	Any valid directory.	The temporary directory to use for document captures and transfers. For example, capture.work.directory=C:/inserver/temp

Group	Setting	Options	Description
Render Settings	render.fileimageservice.url	URL	This setting specifies the base URL for the File Conversion Service. For example, render.fileimageservice.url=http://localhost:1337 You can request the rendering of raster content from Perceptive Content through the GET: /document/{id}/page/{pageID}/rendition Integration Server call. The call retrieves a PNG rendition of a document page. **Note** You must install the File Conversion Service prior to using the Integration Server call. For installation instructions, see the Integration Server 7.1 > File Conversion Service Help.

Appendix: B X.509 LDAP authentication translator

The following table defines the settings for the X.509 LDAP authentication translator plugin. The plugin uses attributes to map a client X.509 certificate to an LDAP user. To map successfully, you must use unique attributes, such as an email address or user ID. If multiple LDAP entries match the X.509 client certificate, then mapping is unsuccessful.

The sso.authenticator.class setting for this plugin is com.imagenow.authentication.translator.x509ldap.X509LDAPAuthenticationTranslator.

Note To use this plugin, you must configure your application server to use SSL/TLS with client certificate authentication.

Setting	Options	Description
sso.x509ldap.ldap.server	Any valid address.	The LDAP server address.
sso.x509ldap.ldap.port	Any valid port number.	Optional. The LDAP server port number. If SSL is enabled, the default is 636. If SSL is not enabled, the default is 389.
sso.x509ldap.max.connections	Any valid number.	Optional. The maximum number of connections that can be made to the LDAP server. The default is 10.
sso.x509ldap.ldap.ssl	TRUE FALSE	Optional. Specifies whether to enable SSL/TLS for connections to the LDAP server. The default is FALSE.
sso.x509ldap.truststore.location	Any valid path.	Path to the truststore to use to validate the LDAP server certificates.
sso.x509ldap.truststore.password	Any valid password.	Optional. The password used to verify the integrity of the truststore.
sso.x509ldap.validate.server.certificates	TRUE FALSE	Optional. Specifies whether LDAP server certificates should be validated when SSL is enabled. The default is TRUE
sso.x509ldap.bind.dn	Any valid name.	The distinguished name of an account to use for LDAP server authentication.
sso.x509ldap.bind.password	Any valid password.	The password to be used for LDAP server authentication.

Setting	Options	Description
sso.x509ldap.anonymous.bind	TRUE FALSE	Optional. Specifies whether to use an anonymous login for LDAP server authentication. The default is FALSE.
sso.x509ldap.base.dn	Any valid name.	The distinguished name of the directory tree to be searched.
sso.x509ldap.x509.mapping.attribute	Any valid name.	The name of the X.509 attribute to be used for mapping client certificates to LDAP users.
sso.x509ldap.ldap.mapping.attribute	Any valid name.	The name of the LDAP attribute to be used for mapping client certificates to LDAP users.
sso.x509ldap.ldap.user.attribute	Any valid name.	The name of the LDAP attribute to use as the user's principal name when a client certificate can be mapped to an LDAP user.
sso.x509ldap.log.level	CONFIG, INFO, WARNING, or SEVERE.	Optional. The log level specifying which message levels will be logged. The default is SEVERE.
sso.x509ldap.log.directory	Any valid directory.	Optional. The directory to where log files will be written. The default is LOGS.

Example configuration

The following is an example of an X.509 LDAP authentication translator plugin configuration.

```
sso.authenticator.class=com.imagenow.authentication.translator.x509ldap.X509LDAPAuthen
ticationTranslator
sso.x509ldap.ldap.server=server.example.org
sso.x509ldap.ldap.ssl=true
sso.x509ldap.truststore.location=truststore.jks
sso.x509ldap.bind.dn=cn=ldapuser,ou=accounts,dc=example,dc=org
sso.x509ldap.bind.password=userPassword
sso.x509ldap.base.dn=ou=people,dc=example,dc=org
sso.x509ldap.x509.mapping.attribute=emailAddress
sso.x509ldap.ldap.mapping.attribute=mail
sso.x509ldap.ldap.user.attribute=sAMAccountName
```