Perceptive Integration Server Single Sign-On Plugin

Advanced Design and Setup Guide

Perceptive Content Version: 7.0.x

Written by: Product Knowledge, R&D
Date: October 2014

Table of Contents

Overview

This guide supplements the Integration Server Single Sign-On Solutions Guide. While most customers use the prepackaged plugins, it is possible to author a plugin that further customizes the validation and identification of the authenticated user.

Plugin Authoring Steps

To author a plugin, complete the following steps.

1. Obtain the **authentication-translator-1.0.0.jar** file from the web directory of an Integration Server deployment or extract it from the Integration Server WAR file. This jar file contains the classes and source required for developing a custom authentication class.

Class	Description
com.imagenow.authentication.translator. AuthenticationTranslator	Abstract base class that defines the interface for all authentication plugins.
com.imagenow.authentication.translator. HTTPHeaderTranslator	A concrete implementation of an authentication plugin provided for reference. This plugin can read the username from an HTTP header with a custom name supplied by the application.

2. Create a java class that derives from **com.imagenow.authentication.translator.AuthenticationTranslator**.
   - Choose a meaningful class and package name to reflect what your plugin does.
   - Your class must implement the following method. If an exception is thrown, the user is denied access to the application. If the return value is the empty string, the user is denied access to the application. If a valid user name is returned, the application attempts to authenticate as that user.

     ```
     public abstract String getUsername(Map<String, List<String>> headers);
     ```

   - Your class can get access to properties defined in the Integration Server configuration file (**integrationserver.ini**) through the `getProperty` method on the base class.

     ```
     String headerName = getProperty("sso.httpheader.name");
     ```

3. Compile your class and bundle it into a JAR file with a name of your choosing.

Plugin Deployment Steps

To deploy a plugin, complete the following steps.

1. Copy the JAR file to the common lib directory of the Tomcat installation where Integration Server is deployed (`%TOMCAT_HOME%/lib`).
2. Update the **sso.authenticator.class** setting in the **WEB-INF/integrationserver.ini** file to reference the fully qualified class name of your custom authenticator class.

   Note If you created customizable properties for your authenticator plugin, you must add the properties to the **integrationserver.ini** file.
3. Restart the Tomcat service.