

Kofax FraudOne

Thin Client Customization Layer

Version: 4.5.0

Date: 2020-11-04

The KOFAX logo is displayed in a bold, blue, sans-serif font. The letters are thick and closely spaced, with a consistent weight throughout the word.

© 2020 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table of Contents

Preface	4
Related documentation.....	4
Training.....	5
Getting help with Kofax products.....	5
Chapter 1: Customization layer	6
Custom configuration.....	6
Business Model.....	6
Thin Client.....	6
Chapter 2: Core overview	7
General settings.....	7
Prior to Release 4.3.....	7
Release 4.3 and higher.....	7
Core contents.....	7
Chapter 3: Custom build	9
Requirements for a customer build.....	9
War / Ear package.....	10
Chapter 4: Environment	11
Chapter 5: Runtime components	12
Chapter 6: Plugin components	14

Preface

This document contains information so that customer customizations can be made on the Thin Client.

Customizations can be performed via settings in the Thin Client properties as prescribed in the 'Configuration Guide' or also in conjunction with customizations performed within the business model. Within the business model the settings made in the custom.zip in some cases affect the Thin Client as well.

Related documentation

The full documentation set for KOfax FraudOne is available at the following location:

<https://docshield.kofax.com/Portal/Products/FO/4.5.0-th2k87ey6r/FO.htm>

In addition to this guide, the documentation set includes the following items:

Guides

- *Kofax FraudOne Administrator's Guide*
- *Kofax FraudOne Data Warehouse Installation and Operation Guide*
- *Kofax FraudOne Extended Reporting Features and Statistics*
- *Kofax FraudOne Feature Codes*
- *Kofax FraudOne Installation and Migration Guide*
- *Kofax FraudOne Java Client Customization Guide*
- *Kofax FraudOne Java Client Customization Layer*
- *Kofax FraudOne License Management*
- *Kofax FraudOne Report Component Installation Guide*
- *Kofax FraudOne SignCheck Result Codes*
- *Kofax FraudOne Standard Reporting Features and Statistics*
- *Kofax FraudOne The Book on CRS*
- *Kofax FraudOne Thin Client Customization Guide*

Interfaces

- *Kofax FraudOne Archive Interface Server*
- *Kofax FraudOne ASV Blackbox*
- *Kofax FraudOne Global Fraud Signature Web Service Developer's Guide*
- *Kofax FraudOne Common API Specifications for GIA Engines*
- *Kofax FraudOne Service Program Interfaces*
- *Kofax FraudOne User Login Procedure*

- *Kofax FraudOne Standard Teller Interface*
- *Kofax FraudOne Variant Cleanup Utility*

Online Help

- *Kofax FraudOne Administration Client Help*
- *Kofax FraudOne Java Client Help*
- *Kofax FraudOne Server Monitor Help*
- *Kofax FraudOne Thin Client Help*

Training

Kofax offers both classroom and computer-based training that will help you make the most of your Kofax FraudOne solution. Visit the Kofax website at www.kofax.com for complete details about the available training options and schedules.

Getting help with Kofax products

The [Kofax Knowledge Base](#) repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the [Kofax website](#) and select **Support** on the home page.

Note The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.
Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.
Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).
Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).
Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.
Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

Chapter 1

Customization layer

The Thin Client contains a customization layer which allows the client to be customized for customer requests.

Custom configuration

The Thin Client can be configured from two location depending on the functions. The custom.zip can be used to configure some functions which are controlled by the business model. Thin Client specific functionality can be configured in the Thin Client.

Business Model

Some functionality of the Thin Client is derived by customization performed in the business model by overwriting core classes and properties with custom properties files. See the document *Kofax FraudOne Java Client Customization Layer* regarding customization of the Java Client.

Thin Client

The Thin Client configuration layer allows customer specific alterations to .java and .jsp files which affect only the current customer. These possibilities are described in this document. These modifications affect the Thin Client only.

Chapter 2

Core overview

For a Thin Client installation the customer receives a .WAR (default name: SignPlus.war) file. The .WAR file contains all core and customer components, property files as well as all required modules for the Thin Client. In order to support a customization layer for the Thin Client the core modules have been separated and are located as with all other core components. The required customized classes, Java Server Pages and properties for the customer must then be used to compliment the core Thin Client. Afterwards the .WAR package must be built containing the core components.

General settings

The core build distributes the data to core/dev without any packaging. The packaging is to be performed when the customer specific modifications are completed and a customer shipment is required. The customer components can basically replace any items within the core data.

Prior to Release 4.3

When developing building a customer package in previous releases all required modifications were performed in customer branches of the Thin Client core. This resulted in all customer modifications being located in branches of the core components. For each release the entire Thin Client could be tagged to reflect the actual customer release.

Release 4.3 and higher

Core contents

The core components for the respective Thin Client release are provided in the ThinClient Core folder (\\gimli.emea.kofax.com\spdfs\dev\rel\core\SignPlus\R44\freeze\tclient2). The core contents include the following items:

Root:

1. Main Java Server pages (.jsp's)
2. Core release .rel and .ver files

3. 'applets' folder
4. 'classes' folder
This contains all core class files which are included in the tclient.jar and is not intended for a release. These subfolders and files are intended only for building a tclient.jar after any required customization has been performed.
5. 'custom' folder
This is an empty folder structure for showing where the customer files are to be located.
6. 'help' folder
Contains the standard help files for the Thin Client
7. 'jsp' folder
Contains further Java Server pages for the Thin Client implementation.
8. 'samples' folder
Can contain any sample files for the Thin Client.
9. 'script' folder
Contains Java Script files required by the Thin Client.
10. 'WEB-INF' folder
Contains all settings, properties and runtimes for the Thin Client.
11. CORE.MANIFEST
The manifest files contains build information regarding the core build.

Chapter 3

Custom build

All thin client deliveries are customer dependent.

Prior to Rel 4.2 required that at least one file (build.properties) must be branched and modified for the customer. The package was then built and distributed to the projects folder for the respective SignPlus release.

For Rel 4.3 and higher the build.properties becomes a core file and a second property file (or similar) is required to build the customer modules and package the customer files with the core files to a customer war file.

Requirements for a customer build

A customer shipment or package is always based on the current or required core package complemented with any class or jsp changes and then the required property files added. Any further customer specific files can be added as required.

The following steps should be performed by a (ant) script for creating the customer package.

1. Core components are all copied to a working folder. This will also be the 'dist' folder.
2. Add all customer class files to the 'classes' folder. The customer files will replace the core class files.
3. If no customer customization was required or performed on any classes then the 'classes' folder can be deleted and the core tclient.jar used as is.

If customization has been performed on one or more class files then a new tclient.jar must be built. The tclient.jar is will be located in the WEB-INF/lib folder and replace the core version already located in the 'lib' folder.

In either case the 'classes' folder must be removed after the correct tclient.jar is located in the WEB-INF/lib folder. 'classes' is not intended for shipments to the customer and should never be in the war or ear file.

4. All jsp files which have been modified for the customer must be copied into the respective location to replace the core version.
5. All property files required for the customer release should be placed in the 'WEB-INF/classes' folder of the 'dist' location.
6. The core .ver and .rel files located in the root path of the core release should remain untouched. These reflect the core version and release.
7. Additionally the .ver file and .rel file from the customer project should also be placed in the root path and should reflect the customer specific modifications for the release.

8. Now the required runtime .jar files are required in the WEB-INF/lib folder. The versions required can be located in the build.properties file located in the core checkout.
See Runtime Components for a sample list of .jar files included in the package.
9. Additionally the runtime binaries must be copied to a new folder, 'bin' located in the WEB-INF/lib folder.
See the sample list of the binary Runtime Files included in the package.
10. The 'bin' folder will also include the folder 'lmdata' which contains the demo license files for SignPlus.
11. Add the Plugin components the package. A sample list of files required can be found in the Plugin Components section.

War / Ear package

The war file is required for deploying to a Tomcat WebServer. If the customer uses WebSphere an ear file will be required.

Along with the package a .ver file containing the core tag and a .rel file which documents alterations to the package both of which should be redistributed with any customer shipments to reflect the current core version installed.

File	Usage
Signplus.war	Customer package for shipping
Signplus.ear	Customer package for shipping
TClient2.rel	Core specific release info
TClient2.ver	Tag from the core release

Chapter 4

Environment

For developing on Rel 4.4 the following components are recommended.

Component	Version	Comments
Eclipse	Mars / Neon	Neon for Tomcat 8.5.x
Java	1.8 JDK	
Tomcat	8.x or 8.5.x	

Chapter 5

Runtime components

WEB-INF/lib	WEB-INF/lib/bin	WEB-INF/lib/bin/ lmdata
Bin ----->		
	lmdata ----->	
Acme.jar	fsxtd.dll	SignPlus2.dat
c3p0-0.9.1.1.jar	fsxtkkrn.dll	SignPlus2.lic
castor-1.0.1-xml.jar	getter.dll*	
commons-beanutils.jar	lfbmp13n.dll	
commons-codec-1.3.jar	LFCMP13n.DLL	
commons-collections.jar	lffax13n.dll	
commons-digester.jar	lfgif13n.dll	
commons-fileupload.jar	lfica13n.dll	
commons-httpclient-3.1.jar	lfimg13n.dll	
commons-lang.jar	lflma13n.dll	
commons-logging-1.1.1.jar	lflmb13n.dll	
commons-validator.jar	lfpcx13n.dll	
custom.jar	Lfpng13n.dll	
custom.ver	lftif13n.dll	
esapi-2.0.1.jar	Lfwmf13n.dll	
ics.rel	lfXbm13n.dll	
icsClient.jar	log4cxx.dll	
jakarta-oro.jar	LTDIS13n.dll	
jclient.rel	ltefx13n.dll	
jclient_custom.rel	ltfil13n.DLL	
jdom.jar	lting13n.dll	
logback-classic-1.1.2.jar	ltkrn13n.dll	
logback-core-1.1.2.jar	littwn13n.dll	
mail-1.4.jar	Lvkrn13n.dll	
service#service.rel	msvcp71.dll	

WEB-INF/lib	WEB-INF/lib/bin	WEB-INF/lib/bin/ lmdata
service#softpro.rel	msvcr71.dll	
service#tiff.rel	MSVCRT.DLL	
service.jar	QueueClt.dll*	
service.ver	QueueSvr.EXE*	
SignWare.rel	SharedCommon.rel	
slf4j-api-1.7.7.jar	SharedCommon.ver	
softpro.jar	sival.rel	
softpro.ver	SPDSV_DLL.dll	
spclient.jar	SPFreedImage_1.dll	
spclient.ver	SPLCSvr.exe*	
spnative.jar	splm2.dll	
SPSignDoc_4.0.jar	splm2.rel	
SPSignware_32.jar	splm2lv432.dll	
standard-1.0.4.jar	splm2view.exe	
struts-menu.jar	SPSignDoc_4.0.dll	
struts.jar	SPSUtils.dll*	
tclient.jar	SPSVRS.dll*	
tiff.jar	SPTiff_31.dll	
tiff.ver	SP_DEC.DLL*	
urlrewritefilter-4.0.4.jar	SP_Dec.ver	
webcastellum-1.8.3.jar	SP_ImgRen.dll	
xercesImpl.jar	SP_Logon.dll*	
XjLog-1.0.9.jar	SP_QtCore4.dll	
	SP_QtGui4.dll	
	SP_SBLib.dll*	
	SP_SbLib.ver	
	SP_SDK_32.dll	
	tiffib.dll*	
	xclogd_1.0.dll	
	xerces-c_3_1.dll	

Chapter 6

Plugin components

download	download/nt
nt ----->	ActiveCAP.cab ActiveISec.cab ActiveSD.cab ActiveTC.cab csXImage.cab csXImage.lpk Plugin.rel plugin.txt Plugin.ver SpCapPlugin.crx SpCapPlugin.jar SpCapPlugin.xpi SpIsecPlugin.crx SpIsecPlugin.jar SpIsecPlugin.xpi SPPlugin.crx SPPlugin.jar SPPlugin.xpi SpSdPlugin.crx SpSdPlugin.jar SpSdPlugin.xpi SPTelSvr.exe