

Kofax FraudOne

Thin Client Customization Guide

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	7
Related documentation.....	7
Training.....	8
Getting help with Kofax products.....	8
Chapter 1: Central configuration	9
Chapter 2: Custom configuration	12
Business Model.....	12
Thin Client.....	12
Chapter 3: General and view settings	14
General settings.....	14
New window.....	14
Right mouse click.....	14
Font selection.....	15
Tooltips.....	15
Drag and Drop.....	15
Bank Name.....	15
Logo.....	16
Log settings.....	16
Login.....	16
Session timeout.....	18
Use ActiveX.....	18
Plugin version.....	19
Thin Client license type.....	19
JDBC connection.....	19
SignBase.....	21
Show signatories (default setting).....	21
Show stock images (default setting).....	21
Load check stock images.....	21
Check stock image details position.....	21
Watermark.....	22
SignInfo.....	23
SignInfo Client.....	23
Show SignInfo (default setting).....	23
Image smoothing.....	23

Image compression factor.....	24
Load images.....	24
Image details position.....	24
Clean images preview.....	24
SignCheck.....	25
Document history.....	25
Image tools.....	25
Show check data.....	25
Show history.....	25
Set view persistence.....	26
Servlet settings.....	26
Chapter 4: Menu settings.....	27
General menu settings.....	27
Menu entries.....	27
About menu.....	27
Help button.....	28
Change password.....	28
Log off.....	29
Shortcuts.....	29
SignBase.....	30
Image tools.....	30
Image view settings.....	30
SignInfo.....	31
SignCheck.....	31
SignCheck status.....	31
SignCheck lists.....	32
SignCheck queues.....	32
SignCheck VBD.....	32
SignCheck search Archive.....	32
SignCheck voucher search.....	33
Chapter 5: Functionality settings.....	34
General settings.....	34
SignBase.....	34
Signature capture.....	34
Replace a signature.....	34
Store variant.....	35
Detail restriction text.....	35
SignInfo.....	35

LDAP (Client) support.....	35
SignCheck.....	36
Physical verification.....	36
STV support.....	37
VPSV support.....	37
ICV support.....	37
SignCheck Archive.....	37
Store global fraud.....	38
Require user confirmation to store a global fraud.....	38
Chapter 6: Teller settings.....	39
Teller mode.....	39
Teller error dialog.....	39
Teller shortcuts.....	39
Strip zeros from request data.....	40
Chapter 7: Visibility settings.....	41
General fields.....	41
Search fields.....	41
Customer details.....	42
Account details.....	42
Signatory details.....	43
Signature details.....	44
Global fraud details.....	44
Rules details.....	44
Signatory list view.....	45
Teller buttons.....	45
SignInfo image view.....	45
CheckStock image view.....	46
ICV specific fields.....	46
AccountGrouping controls.....	46
Chapter 8: EasyExtensions.....	48
Display possibilities.....	48
Separate configuration files.....	48
Supported UI widgets.....	49
Supported dialogs/panels and their Layout.....	51
Configuration of EasyExtensions.....	53
Chapter 9: Building a customer client.....	56
Branching and customizations.....	56
Tagging.....	57

Building..... 57

Preface

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

Customizations can be performed via setting in the Thin Client properties 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 Layer*

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

Central configuration

The Thin Client uses the Central Configuration for Release 4.4 and higher. All core and customer property files are now maintained in the Central Configuration Server and can no longer be loaded from the file system. There are 2 basic types of files referred to within this description as 'elements' since there are not files as such anymore. The element types used are:

- **Common:** These entries are core settings and are not to be modified.
- **Private:** The elements are customer specific settings which can be modified.

The following list shows where the elements originate from a previous installation using property files saved on the file system. To ease configuration the element names have been left the same as the file names previously used.

Thin Client specific elements (Common and Private)

Element	Type	Originates from:
ThinClient.properties	Common	WEB-INF/lib/tclient.jar (de/softpro/signplus/thinclient)
ThinClient_de.properties	Common	WEB-INF/lib/tclient.jar (de/softpro/signplus/thinclient)
ThinClient_en.properties	Common	WEB-INF/lib/tclient.jar (de/softpro/signplus/thinclient)
ThinClient_es.properties	Common	WEB-INF/lib/tclient.jar (de/softpro/signplus/thinclient)
ThinClient.properties	Private	Customer specific text (properties)
ThinClient_de.properties	Private	Customer specific text (properties)
ThinClient_en.properties	Private	Customer specific text (properties)
ThinClient_es.properties	Private	Customer specific text (properties)
custom.properties	Private	WEB-INF/classes/custom.properties
tcustom.properties	Private	WEB-INF/classes/tcustom.properties
visibility.properties	Private	WEB-INF/classes/visibility.properties
tcversion.properties	Common	WEB-INF/classes/tcversion.properties

Required from the Business Model:

Element	Type	Originates from:
action.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
AResctictions.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
bno.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
CIInstructions.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)

Element	Type	Originates from:
comments.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
comments_de.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
comments_en.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
comments_es.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
Currency.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
help.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
help_de.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
help_en_US.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
help_es.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
Instructions.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
kernel.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
language.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
language_de.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
language_en.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
language_en_GB.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
Language_es.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
layout.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
lists.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
menu.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
menuSignInfo.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
Scan.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
SCClient.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
SCClientMenu.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
SCClientMenu_de.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
server.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
settings.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
SignBase.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
signdoc.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
sizes.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
tablayout.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)

Required from the Customizations:

Element	Type	Originates from:
xxx.properties	Private	WEB-INF/lib/custom.zip (de/xxx/resources) Customer specific properties to overwrite or compliment the 'Common' Business Model properties.

Required from the Softpro.jar:

Element	Type	Originates from:
SVCMessages.properties	Common	WEB-INF/lib/softpro.jar

Chapter 2

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 functions can be configured in the thin client property files (tcustom.properties and visibility.properties).

Business Model

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

The custom properties files must be named like the core properties files and are normally provided in a custom.zip archive for the business model which needs to be available in the CLASSPATH when the Thin Client is started. Additionally there is one main configuration file named custom.properties which has to be available outside the custom.zip in the CLASSPATH.

Beside the standard Java properties mechanism, by providing a value to a specific key, there are additional functionalities available for custom SignBase and SignCheck configuration.

SignBase

- Specify value in another properties file by using asterisks forward (*custom*)
- BNO specific properties by using bno.properties configuration
- Classpath lookup of properties (properties file lookup is done in order: 1. core, 2. custom.zip, 3. classpath). These property files are always place in the specific release 'config' folder with the ending 'tclient'.

Caching of properties (PropertyHandler: configuration is hold in HashMap)

SignCheck

- Special configuration

Thin Client

The Thin Client configuration file(s) are also required to provide settings independent of the business model. These settings are located in the files:

tcustom.properties

visibility.properties

web.xml

For more details on tcustom.properties see General and view settings, Menu settings, Functionality settings, Teller settings, and Servlet settings.

These modules are used in the Thin Client only.

Chapter 3

General and view settings

Settings which affect general functionality and visualization. The settings are located in the `tcustom.properties`.

General settings

New window

To open the browser in a new window without any toolbars from the browser use the property:

DisableBrowserToolbars

Example

```
DisableBrowserToolbars=false (default)
```

Default: false

This specifies whether or not the Thin Client should be opened in a new popup window without a tool bar, menu bar, status bar or address bar.

To specify the size of the new window (when `DisableBrowserToolbars = true`) the following properties may also be set to:

```
NewWindowWidth
```

```
NewWindowHeight
```

Right mouse click

The right mouse click functionality for popup menus can be disabled if required.

DisableRightMouseClicked

Example

```
DisableRightMouseClicked=true
```

to deactivate the right mouse click.

Default: false

Font selection

Enables whether the user should be able to select the font size within the application.

EnableFontSelect

Default: true

Tooltips

If not desired the display of tooltips may be disabled.

DisableToolTips

Example

```
DisableToolTips=false
```

to no longer display tooltips.

Default: false

This also affect the SignPlus tooltips for displaying detail information from the signatory list.

Also affected is the display of tooltips in SignCheck for tooltips such as reject reasons. When disabled the reject reasons selection view will appear when the reject button is selected.

In both cases no details or reasons list will appear when the mouse hovers over the respective link.

This must be disabled when the Client – Server connection is protected against Cross Site Reference Forgery.

Drag and Drop

The drag and drop feature in the browser can be disabled. Specifies whether drag-able page objects should be disabled. If set to false, drag-able objects may be moved (via keep pressing the left mouse button) around the page and dropped (e.g. function icons). Signatures can be moved also (e.g. for a visual comparison), but after releasing the left mouse button the signature snaps back to the original position.

DisableDrag

Example

```
DisableDrag=true
```

to disable the drag and drop functionality

Default: false

Bank Name

The property BankApplicationName configures the header of the Thin Client.

If SICE.Enabled=true then the Bank Name is controlled by BankSiceApplicationName.

Default: --

Logo

Whether or not the logo should be displayed in the window header can be controlled with:

displayLogo

Example

```
displayLogo=1
```

Default: true

Additionally the actual image to be displayed as the logo can be specified using:

LogoImageName

Example

```
LogoImageName=/res/logo_fraud1_small.gif
```

Default: --

When in teller mode there is a separate property to control the display of the log.

dislayLogo.Teller

This overrides the 'displayLogo' property when in teller mode.

Log settings

Producing a log file with the specified trace level can be set with the following settings.

LogFile

LogLevel

Default: --

Example

```
LogFile=D:\logfile.log
```

```
LogLevel=1
```

The log file for the business model is set in the custom.properties also with a file and level.

Login

There are various means to login to SignPlus with the Thin Client. The means of login is set in the in the custom.properties. The tcustom.properties from the Thin Client described here are used when the

UseSignSec in the custom.properties is set to '0' which deactivates the SignPlus server authentication. In this case the login credentials and user rights are provided by the Thin Client properties.

Login.User

Login.Password

Default: --

Example

```
Login.User=myName
```

```
Login.Password=myPassowrd
```

If used then the appropriate SignBase and SignCheck user rights must also be assigned. This is accomplished using the following properties:

SBRights – SignBase user rights

SCRights – SignCheck user rights

In addition the user rights are specified per BNO:

SBRights.999 – for all BNO's

SBRights.001 – for BNO 001 only

The user rights SBRights and SCRights contain a decimal number which result from a binary OR of the user rights desired. The value entered is the sum of the decimal values of the desired user rights.

Usage	Right	Binary value	Decimal
SBRights	SB_INQUIRY	0x00000001	1
SBRights	SB_CHANGE	0x00000002	2
SBRights	SB_VERIFY	0x00000004	4
SBRights	SB_ADMIN	0x00000008	8
SBRights	SB_Audit	0x00000010	16
SBRights	SB_STV	0x00000020	32
SBRights	SB_Report	0x00000040	64
SCRights	SC_GROUP1	0x00000001	1
SCRights	SC_GROUP2	0x00000002	2
SCRights	SC_PTSV	0x00000004	4
SCRights	SC_VSM	0x00000008	8
SCRights	SC_VC	0x00000010	16
SCRights	SC_VARIANT	0x00000020	32
SCRights	SC_STATS	0x00000040	64
SCRights	SD_VBD	0x00000080	128

Usage	Right	Binary value	Decimal
SCRights	SD_PRINT	0x00000100	256
SCRights	SD_OPERATOR	0x00000200	512
SCRights	SC_ADMIN	0x00000400	1024
SCRights	SC_INQUIRY	0x00000800	2048
SCRights	SC_GROUP3	0x00001000	4096
SCRights	SC_GROUP4	0x00002000	8192
SCRights	SC_PTSV2	0x00004000	16384
SCRights	SC_CHANGEDOC	0x00008000	32768
SCRights	SC_VIEWDECISIONS	0x00010000	65536

Session timeout

Should a session timeout occur within the Web Server for the current client session then a redirect to the login page should occur. The setting specifies whether or not the user should be redirected to the login page after a session timeout, if not, keep alive requests are sent periodically in teller mode before session timeout occurs.

HandleSessionTimeout

Example

```
HandleSessionTimeout=true
```

Default: true

This feature is disabled (false) in Teller operation mode (when isTeller=true).

Use ActiveX

This setting is required when the Internet Explorer is used as browser for the Thin Client. The ActiveX component for the browser will be downloaded from the server to the client is set and an update or installation of the ActiveX is required.

UseActiveX

Example

```
UseActiveX=true
```

Default: false

If enabled (=true or =1) then the version required may also be specified. This is performed with the following setting.

ActiveXVersion

Example

```
ActiveXVersion= 1,6,5,5
```

Plugin version

If the Mozilla plugin is used the required version can be specified. This is performed with the following settings.

PluginVersion.Major

PluginVersion.Minor

PluginVersion.Revision

PluginVersion.Build

Example

```
PluginVersion.Major=1
```

```
PluginVersion.Minor=6
```

```
PluginVersion.Revision=5
```

```
PluginVersion.Build=5
```

Default: --

Thin Client license type

This is used to specify the license type used for the customer. When using a seat license the PCSig for the current session is obtained from the client via the ActiveX for IE or the plugin for Mozilla, otherwise the PCSig is created by the business model. This should never be changed by the customer or without consent from sales.

LicenseType

Example

```
LicenseType=concurrent
```

Default: seat

JDBC connection

To provide a JDBC connection for components such as SignArchive which reads archive document directly via JDBS there are properties provided for this configuration. Depending on which database is used the properties differ.

To define the driver used for the installed engine set:

Engine Property=Sample

Database	Engine	Engine Property	Sample
DB2	UDB	JDBCClass.DB2UDB	com.ibm.db2.jcc.DB2Driver
	ZOS	JDBCClass.DB2ZOS	com.ibm.db2.jcc.DB2Driver
Oracle	OCI	JDBCClass.ORACLE-OCI	oracle.jdbc.OracleDriver
		JDBCClass.ORACLE8-OCI	oracle.jdbc.OracleDriver
	Thin	JDBCClass.ORACLE-THIN	oracle.jdbc.OracleDriver
Microsoft	SQL	JDBCClass.MSSQL	com.microsoft.sqlserver.jdbc.SQLServerDriver (*)
SyBase		JDBCClass.SYBASE	com.sybase.jdbc2.jdbc.SybDriver
		JDBCClass.SYBASE-DEF	sybase.asejdbc.ASEDriver

Note Prior to MSSQL 2005 requires: com.microsoft.jdbc.sqlserver.SQLServerDriver

Prior to MSSQL 2005 requires: com.microsoft.jdbc.sqlserver.SQLServerDriver

URL definition

To define the driver used for the installed engine set:

URL Property=Sample

Database	Engine	URL Property	Sample
DB2	UDB	JDBCURL1.DB2UDB	jdbc:db2:%(d)
	ZOS	JDBCURL1.DB2ZOS	jdbc:db2:%(d)
Oracle	OCI	JDBCURL1.ORACLE-OCI	jdbc:oracle:oci:@%(d)
		JDBCURL1.ORACLE8-OCI	jdbc:oracle:oci8:@%(d)
	Thin	JDBCURL1.ORACLE-THIN	jdbc:oracle:thin:@%(s):1521:%(d)
Microsoft	SQL	JDBCURL1.MSSQL	jdbc:microsoft:sqlserver://%(s); DatabaseName=%(d); SelectMethod=Cursor
SyBase		JDBCURL1.SYBASE	jdbc:default:connection? BE_AS_JDBC_COMPLIANT_AS_POSSIBLE=true& PROTOCOL_CAPTURE=SYBASE.TRC
		JDBCURL1.SYBASE-DEF	jdbc:sybase:Tds:%(s):5000/%(d)? BE_AS_JDBC_COMPLIANT_AS_POSSIBLE=true& PROTOCOL_CAPTURE=SYBASE.TRC

These items will be filled internally with:

%(d) will be replaced during runtime with the value from entry SCArchive.Connect.DBName.

%(s) will be replaced during runtime with the value from entry SCArchive.Connect.DBServer.

SignBase

Show signatories (default setting)

Set the default value for the SignBase view to be used. After enabling this option, the list of all signatories belonging to the loaded customer / account is displayed by default as a search result in SignBase.

DefaultShowSignatory

Example

```
DefaultShowSignatory=1
```

Default: true

Show stock images (default setting)

Set the default value for the SignBase view to be used. This has an affect only if 'Checkstock=1' is set in the custom.properties.

DefaultShowStockImage

Example

```
DefaultShowStockImage=1
```

Default: true

Load check stock images

Set the number of check stock images to be loaded. All further images will be loaded on demand as required.

LimitCheckStock

Example

```
LimitCheckStock=5
```

Check stock image details position

This parameter applies to the check stock images. The property specifies whether image details should be shown on the right of the image instead of below the image. This is useful in situations where the screen resolution is small. Setting the value to true will result in image details being displayed to the right of the image. Setting the value to false will result in the image details being displayed below the image.

CheckStockShowDetailsOnRight

Example

`CheckStockShowDetailsOnRight=true`

Default: true

Watermark

This parameter applies to the signatures, account images and check stock images. When enabled a watermark will be placed on the images shown in SignBase and the SignBase view in SignCheck. The Watermark is always a text entry and can be configured as described in the following properties.

EnableWatermark

Example

`EnableWatermark=true`

Default: true

Text for the watermark

Configurable watermark text to be displayed.

WatermarkText

Example

`WatermarkText=My Text`

Default: Kofax FraudOne

Font for the watermark

Font size for the watermark

Font bold for the watermark

Insert the watermark using 'Bold' if specified.

WatermarkFontBold

Example

`WatermarkFontBold=true`

Default: false

Font italic for the watermark

Opacity for the watermark

SignInfo

SignInfo Client

If the Thin Client is to be used as a SignInfo client then this setting should be set to true.

SICE.Enabled

Example

```
SICE.Enabled=true
```

Default: false

If SICE.Enabled=true then the Bank Name will be set by the BankSiceApplicationName property.

Show SignInfo (default setting)

Set the default value for the SignBase view to be used. This has an affect only if 'SignInfo=1' is set in the custom.properties.

DefaultShowSignInfo

Example

```
DefaultShowSignInfo=1
```

Default: false

Image smoothing

This property specifies whether or not images are smoothed before being sent from the server to the browser. Setting this property to 1 will result in better quality images but will slow down image transfer and processing. Setting this property to 0 will result in better performance, but poorer quality images.

ImageSmoothing

Example

```
ImageSmoothing=1
```

Default: true

Image compression factor

In order to set the best ratio between image quality and transmission time the JPEG Compression factor in the range from 0.0 to 1.0:

0.0 - lowest Quality

1.0 - highest Quality

jpegQuality

Example

```
jpegQuality=0.4
```

Default: 0.4

Load images

Set the number of images to be loaded. All further images will be loaded on demand as required.

LimitSignInfo

Example

```
LimitSignInfo=5
```

Image details position

This parameter applies to the images displayed in the SICE application mode, and to the account images displayed as reference data in SignCheck. The property specifies whether image details should be shown on the right of the image instead of below the image. This is useful in situations where the screen resolution is small. Setting the value to true will result in image details being displayed to the right of the image. Setting the value to false will result in the image details being displayed below the image.

SignInfoShowDetailsOnRight

Example

```
SignInfoShowDetailsOnRight=true
```

Default: true

Clean images preview

In the SignInfo image view, hovering over the tree entries displays a preview of already loaded images. This property specifies whether or not the preview image should be “smoothed” before being sent to the browser. Setting this value to false will result in faster loading of the preview images but lower quality. Setting the value to true will result in better quality preview images, but slower performance.

SignInfoCleanImagePreview

Example

```
SignInfoCleanImagePreview=false
```

Default: false

SignCheck

Document history

If set to false only the history of the workflow is displayed, otherwise the full history is displayed (including APIA-results).

EnableFullDocumentHistory

Example

```
EnableFullDocumentHistory=true
```

Default: true

Image tools

Disables the image tools in SignCheck Clients i.e. Image transformation tools such as rotate, clean, invert in SignCheck Clients.

SignCheckDisableImageTools

Example

```
SignCheckDisableImageTools=false
```

Default: false

Show check data

Specify whether or not the SignCheck document data should be displayed by default when a document is loaded.

SignCheckShowCheckData

Example

```
SignCheckShowCheckData=false
```

Default: true

Show history

Set to true if item result history should be displayed in SignCheck view by default.

SignCheckShowHistoryData

Example

`SignCheckShowHistoryData=true`

Default: true

Set view persistence

When enabled the user settings for the current views is saved for further documents if this behavior has been configured. The views settings such as front, back, etc. appear then as with the previous document(s). The setting is valid for the current session.

EnableSignCheckViewPersistence

Example

`EnableSignCheckViewPersistence=true`

Default: true

Servlet settings

The only servlet setting which can be changed is the setting for the InstanceName of the servlet when started. This provides the capability to create property files for this web server only. The Name is always 'InstanceName' and the value should be set to match the Instance Name set in the Administration Client.

The setting should follow the <web-app>... setting. An entry may already be in your web.xml file (commented out).

```
<web-app .....>
...
  <context-param>
    <param-name>InstanceName</param-name>
    <param-value>MyServerInstanceName</param-value>
  </context-param>
...
</web-app>
```

Chapter 4

Menu settings

Some settings are available which control the visualization of menus. The settings are located in the `tcustom.properties`.

General menu settings

Menu entries

If desired it is possible to disable all menu entries. The setting `true` disables all menu entries and overrides any other settings.

DisableAllMenuEntries

Example

```
DisableAllMenuEntries=true
```

disables all menus

Default: `false`

In addition the about menu can be controlled separately in teller mode. The settings for the teller mode override the 'DisableAllMenuEntries' setting when in Teller mode. The supported teller settings are:

```
DisableAllMenuEntries.Teller
```

```
DisableAllMenuEntries.Teller.Kld2Teller
```

```
DisableAllMenuEntries.Teller.Ups2Teller
```

About menu

The 'About' menu can be turned on/off as desired.

EnableAboutMenu

Example

```
EnableAboutMenu=true
```

displays the about menu.

Default: true

In addition the about menu can be controlled separately in teller mode. The settings for the teller mode override the 'EnableAboutMenu' setting when in Teller mode. The supported teller settings are:

`EnableAboutMenu.Teller`

`EnableAboutMenu.Teller.Kld2Teller`

`EnableAboutMenu.Teller.Ups2Teller`

Help button

The 'Help' functionality can be turned on/off as desired.

DisableHelpButton

Example

`DisableHelpButton=true`

turns the help button off.

Default: false

In addition the help button can be controlled separately in teller mode. The settings for the teller mode override the 'DisableHelpButton' setting when in Teller mode. The supported teller settings are:

`DisableHelpButton.Teller`

`DisableHelpButton.Teller.Kld2Teller`

`DisableHelpButton.Teller.Ups2Teller`

Change password

Specifies whether or not password changing is enabled. If set to 1 the "Change password" menu entry will be available and this functionality will be enabled. Setting the property to 0 will disable password changing for all users.

ChangePwd

Example

`ChangePwd=false`

prohibits user from changing their password in the Thin Client.

This setting is only considered if `UseSignSec=1` is set in the `custom.properties` file.

Default: true

Log off

Specifies whether or not the Logoff option should be displayed in the menu bar.

EnableLogoffMenu

Example

```
EnableLogoffMenu=false
```

to remove the 'Log off' menu.

Default: true

Shortcuts

Shortcuts can be specified for various functions. The functions are predefined and can be associated with a key of combination of keys.

First of all the shortcuts can be activated or deactivated using the property:

EnableThinClientShortcuts

Example

```
EnableThinClientShortcuts=true
```

activates the specified shortcuts if DisableBrowserShortcuts is not set to true.

Default: true

Functions which are supported and can have a key or key combination assigned:

ShortcutOpenHelp

ShortcutSearchAccount

ShortcutSearchCustomer

ShortcutAcceptGetNext

ShortcutRejectGetNext

ShortcutHoldGetNext

ShortcutGetNext

ShortcutDocumentHistory

Each function key that should be configured may consist of:

+, -, F1-F12, A-Z, 0-9, INS and one of the additional keys SHIFT, CTRL and ALT.

Sample of some settings allowed: F3, SHIFT+F3

Another property which affects the shortcuts functionality is:

```
DisableBrowserShortcuts
```

Specifies whether or not shortcuts (browser shortcuts and Thin Client shortcuts) should be disabled. All shortcuts which should be disabled needs to be listed and separated by blanks as values for additional property entry BrowserShortcuts.

SignBase

Image tools

If desired the image tools in SignBase Clients i.e. Image transformation tools such as rotate, clean, invert in SignBase Clients.

DisableImageTools

Example

```
DisableImageTools=true
```

to no longer show the image tools.

Default: false

Default: true

Image view settings

Images view settings can be configured with the 'MaxWidth', 'MaxHeight' and 'ScalingStrategy' to be displayed with the Thin Client. These settings are available for the views listed below and have the following meaning:

MaxWidth - property specifies the maximum width in the given context,

MaxHeight - property specifies the maximum height in the given context

ScalingStrategy - property specifies the strategy to be used for scaling in the given context.

Possible values are:

.ScalingStrategy = 0 -> scale to width and height of browser window (default)

.ScalingStrategy = 1 -> scale to width of browser window

.ScalingStrategy = 2 -> scale to height of browser window

Strategy 0 is recommended where images are mixed (both portrait and landscape)

Strategy 1 is recommended for images that are mainly portrait (i.e. width is greater than height)

Strategy 2 is recommended for images that are mainly landscape (i.e. height is greater than width)

The settings can be made for the following views:

`SignatoryListView`

`ComplexRulesView`

`SignatoryDetailsView`

`SignWithShortList`

`ImageView`

`SignInfoImagePreview`

`VpsvView`

`SignCheckDocumentFront`

`SignCheckDocumentBack`

`SignCheckDocumentSignature`

`StoreVariantView`

`SiceResultList`

`CheckStockResultList`

`CheckStockResultListPreview`

Example

`SignatoryListView.MaxWidth`

`SignatoryListView.MaxHeight`

`SignatoryListView.ScalingStrategy`

SignInfo

There are presently no SignInfo specific menu settings.

SignCheck

SignCheck status

Enables the menu to search by document status in SignCheck.

EnableSignCheckStatus

Example

```
EnableSignCheckStatus=true
```

Default: true

SignCheck lists

Enable the menu to search by lists in SignCheck.

EnableSignCheckLists

Example

```
EnableSignCheckLists=true
```

Default: true

SignCheck queues

Enable the menu to search and process by the selected Queue.

EnableSignCheckQueues

Example

```
EnableSignCheckQueues=true
```

Default: true

SignCheck VBD

Enable the menu to search for a specific document from the VBD queue

EnableSignCheckVBD

Example

```
EnableSignCheckVBD=true
```

Default: false

SignCheck search Archive

Enable the menu to search in the Archive for documents.

EnableSignCheckArchive

Example

```
EnableSignCheckArchive=true
```


Default: false

SignCheck voucher search

Enable the functionality to search for a voucher in SignCheck

EnableSignCheckSearch

Example

```
EnableSignCheckSearch=true
```

Default: false

Chapter 5

Functionality settings

These functions offer the selection of various SignBase and SignCheck functionality within the Thin Client.

General settings

There are presently no general functionality settings configurable. The settings are located in the `tcustom.properties`.

SignBase

Signature capture

Enables the capability to capture signatures within SignBase. If activated the user can capture a signature to save in SignBase for further usage. This signature capturing is performed via a SignPad.

EnableSignatureCapture

Example

```
EnableSignatureCapture=true
```

Default: true

Replace a signature

Enables the capability to replace an existing signature in SignBase. If activated the user can capture a signature to replace an existing one in SignBase for further usage. This signature capturing is performed via a SignPad.

EnableSignatureReplace

Example

```
EnableSignatureReplace=false
```

Default: true

Store variant

Allows signatures from a SignCheck document to be stored in SignBase as a variant.

EnableStoreAsVariant

Example

```
EnableStoreAsVariant=true
```

Default: true

Detail restriction text

Allow the user to view the restriction text for the account and/or customer object. Set to true if specific account or customer details should be displayable via a "show" button.

EnableCurrentObjectDetails

Example

```
EnableCurrentObjectDetails=false
```

Default: true

SignInfo

LDAP (Client) support

To use client LDAP login within the Thin Client. This functionality should no longer be required for Release 4.3 and higher. The LDAP login is provided on the SignPlus server and should be used there where possible. This may be deprecated at a later time.

UseLDAP

Example

```
UseLDAP=false
```

Default: false

If used (UseLDAP= true) on the client the following settings must be configured as required:

Context:

Set LDAP context class which is used.

Default: com.sun.jndi.ldap.LdapCtxFactory

LDAP.Context

Example

```
LDAP.Context=com.sun.jndi.ldap.LdapCtxFactory
```

Host:

Set IP for the LDAP server to connect to. Default=127.0.0.1

LDAP.Host

Example

```
LDAP.Host=127.0.0.1
```

Port:

Set port to connect to the LDAP server - standard port for LDAP connections is 389.

LDAP.Port

Example

```
LDAP.Port=389
```

DN:

Distinguished name (DN) which describes the attributes in the LDAP tree down to the searched ObjectClass for the user.

information.

LDAP.DN

Example

```
LDAP.DN=host=tsys,o=somebank,c=us,cn=localhost
```

ObjectClass

Objectclass name under which the user information is stored in the LDAP tree.

LDAP.Objectclass

Example

```
LDAP.Objectclass=tssacid
```

SignCheck

Physical verification

Used to enable the physical verification of documents within the inwards clearing queue.

EnablePhysicalVerification

Example

```
EnablePhysicalVerification=true
```

Default: false

STV support

Allows the SignTeller verification view. Valid only in teller mode.

UseSTV

Example

```
UseSTV=true
```

Default: false

VPSV support

Used when the VPSV should be used as the default pad verification process for SignTeller Verification STV.

UseVpsv

Example

```
UseVpsv=true
```

Default: false

ICV support

Set to true if ICV (Day One Verification) should be enabled for the user.

UseICV

Example

```
UseICV=true
```

Default: false

SignCheck Archive

For connecting to SC Archive the following properties are required. The JDBC connection must also be configured.

The properties are configured by:

```
Property=Sample
```

Property	Sample		
	DB2	Oracle	MS SQL
SCArchive.Connect.DBMS	DB2UDB	ORACLE-THIN	MSSQL
SCArchive.Connect.DBName	FRAUD1DW		
SCArchive.Connect.DBServer	ahoR42:1433	ahoR42:1433	ahoR42:1433
SCArchive.Connect.Schema	dwsp		
SCArchive.Connect.User	Signplus		
SCArchive.Connect.PW	signplus		
SCArchive.Image.Impl	de.softpro.signplus.thinclient.signcheck.SCArchive		
SCArchive.Resultlist.MaxRows	50		

Store global fraud

Allows signatures from a SignCheck document snippet to be stored in SignBase as a global fraud in the blacklist.

EnableStoreAsGlobalFraud

Example

```
EnableStoreAsGlobalFraud=true
```

Default: true

Require user confirmation to store a global fraud

Requires a confirmation when storing a global fraud signature in the blacklist. Since the user cannot delete a global fraud signature it is recommended that a user confirmation be required (=true). This setting is relevant only when 'EnableStoreAsGlobalFraud' is set

UserConfirmationForStoreAsGlobalFraud

Example

```
UserConfirmationForStoreAsGlobalFraud=true
```

Default: true

Chapter 6

Teller settings

The teller configuration is independent of SignBase, SignInfo and SignCheck settings. The settings are located in the tcustom.properties.

Teller mode

If the Thin Client is to be used in teller mode then this property should be set to true. If done the client will operate in teller mode only.

IsTeller

Example

```
IsTeller=true
```

Default: false

Teller error dialog

Enable the possibility to show error messages when the Thin Client is operating in teller mode. Determines whether or not error dialog should be displayed after a teller request has failed. The error is reported to the calling application (ReturnTo parameter) if set to false.

EnableTellerErrorDialog

Example

```
EnableTellerErrorDialog=true
```

Default: true

Teller shortcuts

Specifies whether or not shortcuts are enabled in the teller mode.

EnableTellerShortcuts

Example

```
EnableTellerShortcuts=true
```

Default: true

When set to true, the shortcuts can be configured according to the following property names:

```
ShortcutTellerAccept=SHIFT +
```

```
ShortcutTellerReject=SHIFT -
```

Each shortcut can be configured with +, -, F1-F12, A-Z, 0-9, INS and additional with SHIFT, CTRL, ALT.

Strip zeros from request data

When sending a request via the teller interface the leading zeros ('0') from the request before processing in the thin Client. This can be specified for the customer number and / or the account number.

Teller.Search.RemoveZeros.AccountNo

Teller.Search.RemoveZeros.CustomerNo

Example

```
Teller.Search.RemoveZeros.AccountNo=false
```

```
Teller.Search.RemoveZeros.CustomerNo=false
```

Default: false

Chapter 7

Visibility settings

The visualization of many items can be turned on/off to allow customizing of the view panels. All settings described in this chapter are specified in the visibility.properties file and have a default setting 'true' ('1').

For all settings in this chapter:

Settings supported:

Visible: '1' or 'true'

Not visible: '0' or 'false'

Default: true

General fields

Some settings affect multiple views and must be set only once:

BNO.Visible

CountryId.Visible

Bankcode.Visible

BranchNo.Visible

AccountNo.Visible

NotForASV.Visible

VIP.Visible

NotVerified.Visible

Search fields

Fields shown for searching for customer or account:

Search.BNO.Visible

Search.CountryId.Visible

Search.Bankcode.Visible

Search.ClosedAccounts.Visible

Search.AccountNo.Visible

Search.AccountNo2.Visible

Search.AccountDesc.Visible

Search.CustomerName.Visible

Search.S-Lastname.Visible

Search.S-Firstname.Visible

Search.PersonalId.Visible

Search.Date.Visible

Customer details

Fields configurable for the customer details.

Values displayed in the header:

CustomerNo.Visible

CustomerName.Visible

Values displayed in the customer details:

C-CustomerName.Visible

C-CustomerType.Visible

C-Restriction.Visible

C-RestrictionText.Visible

C-RestrictionDate.Visible

C-DateChanged.Visible

C-CustomerSince.Visible

Account details

Fields configurable for the account details:

A-BIC.Visible

A-AccountNo2.Visible
A-AccountDesc.Visible
A-AccountClosed.Visible
A-Restriction.Visible
A-RestrictionText.Visible
A-RestrictionDate.Visible
A-FreeText.Visible
A-Currency.Visible
A-AccountOpened.Visible
A-ZipCode.Visible

Signatory details

Fields configurable for the signatory details:

S-Lastname.Visible
S-Middlename.Visible
S-Firstname.Visible
S-Birthdate.Visible
S-PersonalId.Visible
S-Title.Visible
S-Position.Visible
S-SignatoryRole.Visible
S-Instruction.Visible
S-InstructionValid.Visible
S-Groups.Visible
S-Mandates.Visible
S-CounterVisible.Visible

Signature details

Fields configurable for the signature details:

S-ScannedFrom.Visible

S-ScanDate.Visible

S-ScannedAs.Visible

S-Image.Visible

S-SignDate.Visible

S-SignatoryStyle.Visible

S-SignatoryType.Visible

S-Quality.Visible

Global fraud details

Fields configurable for the global fraud pop-up details.

GF-Source.Visible

GF-ScanDate.Visible

GF-ScannedAs.Visible

Rules details

Fields configurable for the rules details:

M-Right.Visible

M-Currency.Visible

M-MinAmount.Visible

M-Amount.Visible

M-LimitFrom.Visible

M-LimitTo.Visible

M-Who.Visible

M-WithWhom.Visible

Signatory list view

Fields configurable for the signatory list:

L-Info.Visible

L-SignatoryRole.Visible

L-Groups.Visible

L-Style.Visible

L-Quality.Visible

L-Type.Visible

L-Instruction.Visible

L-Complex.Visible

L-Mandates.Visible

L-Signature.Visible

Global fraud settings within the signatory list:

L-GF-Source.Visible

L-GF-ScanDate.Visible

Teller buttons

Fields configurable for teller usage:

TellerButtons.Visible

TellerBackButton.Visible

TellerBackAcceptButton.Visible

TellerBackRejectButton.Visible

TellerCloseButton.Visible

SignInfo image view

Fields configurable for the SignInfo image details:

I-ImagId.Visible

I-Text1.Visible

I-Text2.Visible

I-Scanned.Visible

I-LastChanged.Visible

I-Processed.Visible

I-Remark.Visible

CheckStock image view

Fields configurable for the CheckStock image details:

CheckStockInstructionUID.Visible

CheckStockInstructionCode.Visible

ICV specific fields

Fields configurable for ICV image details:

ICV.BNO.Visible

ICV.CountryId.Visible

ICV.Bankcode.Visible

ICV.Amount.Visible

ICV.Currency.Visible

ICV.Date.Visible

ICV.TransactionNo.Visible

AccountGrouping controls

Fields configurable for the function buttons used for account grouping:

 AG.Accept.All.Visible

 AG.Accept.Undecided.Visible

 AG.Reject.All.Visible

 AG.Reject.Undecided.Visible

-  AG.OnHold.All.Visible
-  AG.OnHold.Undecided.Visible
-  AG.ResetToReady.All.Visible
-  AG.ResetToReady.Undecided.Visible
-  AG.SendAllToServer.Visible
-  AG.SendDecidedToServer.Visible
-  AG.ResetAllToPending.Visible

Chapter 8

EasyExtensions

EasyExtensions is a configuration capability which is performed solely within the custom.zip. The extension data is displayed following the SignPlus data pertinent to the current data source (Account, Customer, Signatory, etc.). Normally this is visible in the detail view(s) of the data.

Should displaying extensions in the Thin Client, not require any specific positioning they can be activated using the EasyExtensions configuration. If the positioning of the extension field to be displayed can't be displayed as desired by the EasyExtensions then further customization will be required. For the user or PS person this requires deeper knowledge of the Thin Client programming.

Display possibilities

The extensions which are displayed using EasyExtensions will be displayed as follows:

- Always following the standard data.
- All extensions display label and a value.
- Values which can be displayed are Text, Text area, Checkbox or Combo box (displayed as a single text entry).
- One label/value pair per row.
- The label is shown in 'Bold'.

For EasyExtensions the configuration of the extensions display is performed as with the Java Client. The Thin Client has no specific customization for the EasyExtensions. For further information regarding the configuration required see *Kofax FraudOne Java Client Customization Guide*.

Separate configuration files

Should a different configuration be required for the Thin Client as opposed to the Java Client then the configuration files used can be extracted from the custom.zip and saved with the additional extension '.tclient'. The files affected are:

- SignBase.properties (SignBase.properties.tclient)
- lists.properties (lists.properties.tclient)
- language.properties (language.properties.tclient)
- The language.properties for customer specific language files (ie. language_de.properties) are also processed.

These files will automatically be packaged in the Thin Client .war (or .ear) file and installed as required.

Supported UI widgets

The supported UI widgets for EasyExtensions are text fields, text areas and checkboxes. The UI widget which is used to display an extension can be configured for each view. The concrete display or edit behavior is triggered by the corresponding extension type. Data from a Combo box will be supported but will always be displayed as a text field with the text of the selected entry. The property entries should be made as with the Java Client for the combo box. This is required in order to show the selected text.



All fields are disabled and view only.

Please note that the separate extension panel is only visible when EasyExtensions are configured for the corresponding dialog.

Text field

Extension Type	Notes
1 (Char)	[Not recommended for text field] Only one character can be displayed/entered
2 (Char-Binary)	[Not recommended for text field] Only one character can be displayed/entered
3 (SmallInt)	Only short values can be displayed/entered
4 (Integer)	Only integer values can be displayed/entered
5 (Decimal)	Only decimal values can be displayed/entered
6 (Date)	Only date values can be displayed/entered. The date format is taken from configuration entry Date.Mask in file language.properties.
7 (String)	

Text area

Extension Type	Notes
1 (Char)	[Not recommended for text area] Only one character can be displayed/entered
2 (Char-Binary)	[Not recommended for text area] Only one character can be displayed/entered

Extension Type	Notes
3 (SmallInt)	[Not recommended for text area] Only short values can be displayed/entered
4 (Integer)	[Not recommended for text area] Only integer values can be displayed/entered
5 (Decimal)	[Not recommended for text area] Only float or double values can be displayed/entered
6 (Date)	[Not recommended for text area] Only date values can be displayed/entered. The date format is taken from configuration entry Date.Mask in file language.properties.
7 (String)	

Combo box

The combo box will be converted to a text field with which contains the selected text. It will not be displayed as a combo box. When using a combo box to display an extension the extension value is expected to be a key.

Extension Type	Notes
1 (Char)	
2 (Char-Binary)	[Not supported for combo box]
3 (SmallInt)	
4 (Integer)	
5 (Decimal)	[Not supported for combo box]
6 (Date)	[Not supported for combo box]
7 (String)	

Checkbox

Extension Type	Notes
1 (Char)	Checked: 't', 'y', '1' (case insensitive) Unchecked: every other value
2 (Char-Binary)	Checked: 't', 'y', '1' (case insensitive) Unchecked: every other value
3 (SmallInt)	Checked: 1 Unchecked: every other value
4 (Integer)	Checked: 1 Unchecked: every other value
5 (Decimal)	[Not supported for checkbox]
6 (Date)	[Not supported for checkbox]

Extension Type	Notes
7 (String)	Checked: "true", "yes", "1" (case insensitive) Unchecked: every other value

Supported dialogs/panels and their Layout

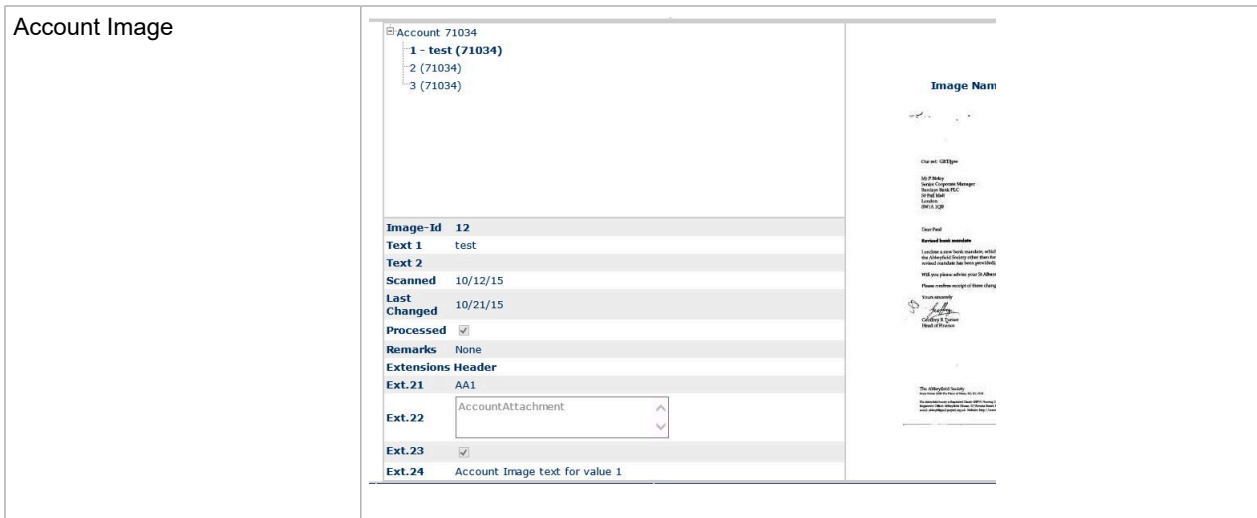
EasyExtensions will be only supported in the specific dialogs or panels which are listed here. If not explicitly mentioned the corresponding dialog/panel supports the displaying of EasyExtensions.

Supports extensions from account, customer, rules, signatories, account images and check stock images.

The following figures shown the existing views of from account, customer, rules, signatures and signatories in Thin Client.

The following figures show the new views of the detail views in the Thin Client. Each screen shot displays a text field, text area, checkbox and combo box (as text field) entry.

<p>Account Details</p>	<p>The screenshot displays the 'Account Details' form with the following fields and values:</p> <ul style="list-style-type: none"> Bank-No.: 001, Country: 840, Bank Code: 31100092 Account-No.: 71034, Restriction: No restriction Branch-No., BIC, Valid until, and Restriction Text: (Empty) Currency: USD Remark: (Text area) Account closed: <input type="checkbox"/> (unchecked) Account opened: <input type="checkbox"/> (unchecked) not for ASV: <input type="checkbox"/> (unchecked) Zip Code: (Text field) Extensions Header: <ul style="list-style-type: none"> Ext.1: Acc Ext.2: Acc Looooong Ext.3: <input checked="" type="checkbox"/> (checked) Ext.4: Account text for value 3
<p>Customer Details</p>	



Configuration of EasyExtensions

EasyExtensions will be configured in SignBase.properties for each dialog. The configuration used for the Java Client will also be used for the Thin Client. Should there be any differences between the two Clients a second signbase.properties file will be packaged with the Thin Client containing the required modifications.

The current extension configuration handling (ExtensionSpecification) is the base for EasyExtensions and remains untouched.

```
ExtensionSpecification.Extensions = ExtName1 ExtName2 ExtName3 ExtName4
ExtensionSpecification.ExtName1.ID = 1
ExtensionSpecification.ExtName1.Type = 7
ExtensionSpecification.ExtName1.Relation = C
ExtensionSpecification.ExtName1.Default =
ExtensionSpecification.ExtName2.ID = 2
ExtensionSpecification.ExtName2.Type = 7
ExtensionSpecification.ExtName2.Relation = C
ExtensionSpecification.ExtName2.Default =
ExtensionSpecification.ExtName3.ID = 3
ExtensionSpecification.ExtName3.Type = 3
ExtensionSpecification.ExtName3.Relation = C
ExtensionSpecification.ExtName3.Default =
ExtensionSpecification.ExtName4.ID = 4
ExtensionSpecification.ExtName4.Type = 1
ExtensionSpecification.ExtName4.Relation = C
ExtensionSpecification.ExtName4.Default =
```

To turn these extensions into EasyExtensions which are displayed for example in the SBAM_Customer dialog the SignBase.properties configuration looks like this:

```
# This setting specifies the extensions which will be used in the dialog
# and also their display order.
SBAM_Customer.Extensions = ExtName1 ExtName2 ExtName3 ExtName4
# This setting specifies the visibility of the extension border.
SBAM_Customer.Extensions.Border.Visible = 1

# This setting specifies the title of the extension border.
# In order to display a border without title specify an empty value here.
```

```

SBAM_Customer.Extensions.Border.Title = *language*extensions.border.title
# This setting overwrites the default height of the extension panel.
SBAM_Customer.Extensions.Dimension.Height =
# This setting overwrites the default width of the extension panel.
SBAM_Customer.Extensions.Dimension.Width =

# Configuration for text field ExtName1
SBAM_Customer.ExtName1.Label = *language*extName1.Label
SBAM_Customer.ExtName1.Size = *sizes*customer.extName1.size
SBAM_Customer.ExtName1.MinSize = *sizes*customer.extName1.minsize
SBAM_Customer.ExtName1.Visible = 1
SBAM_Customer.ExtName1.Enabled = 1
SBAM_Customer.ExtName1.Mandatory = 1
SBAM_Customer.ExtName1.Mask =
SBAM_Customer.ExtName1.AutomaticFocusTraversal =
SBAM_Customer.ExtName1.ConvertToUpperCase =
# This setting specifies which UI widget is used to display the extension.
# Valid values are:
# 1 - text field
# 2 - text area
# 3 - combo box
# 4 - checkbox
SBAM_Customer.ExtName1.Widget = 1

# Configuration for text area ExtName2
SBAM_Customer.ExtName2.Label = *language*extName2.Label
SBAM_Customer.ExtName2.Size = *sizes*customer.extName2.size
SBAM_Customer.ExtName2.MinSize = *sizes*customer.extName2.minsize
SBAM_Customer.ExtName2.Visible = 1
SBAM_Customer.ExtName2.Enabled = 1
SBAM_Customer.ExtName2.Mandatory = 0
SBAM_Customer.ExtName2.Widget = 2
# Configuration for checkbox ExtName3
SBAM_Customer.ExtName3.Label = *language*extName3.Label
SBAM_Customer.ExtName3.Size = 1
SBAM_Customer.ExtName3.MinSize = 1
SBAM_Customer.ExtName3.Visible = 1
SBAM_Customer.ExtName3.Enabled = 1
SBAM_Customer.ExtName3.Mandatory = 0
SBAM_Customer.ExtName3.Widget = 4
# Configuration for combo box ExtName4
SBAM_Customer.ExtName4.Label = *language*extName4.Label
SBAM_Customer.ExtName4.Size = *sizes*customer.extName4.size
SBAM_Customer.ExtName4.MinSize = 0
SBAM_Customer.ExtName4.Visible = 1
SBAM_Customer.ExtName4.Enabled = 1
SBAM_Customer.ExtName4.Mandatory = 0
SBAM_Customer.ExtName4.Widget = 3
# This setting is only available for combo box widgets.
# It is read in the dialog class to build the corresponding combo box
# model.
SBAM_Customer.ExtName4.KeyValueList = *lists*customer.extName4.keyValueList

```

The italic entries are required for the Java Client only. They may be present but will have no effect on the view or functionality provided in the Thin Client.

Example lists.properties configuration for ExtName4 extension:

```

customer.extName4.keyValueList = Value1 Value2 Value3
customer.extName4.keyValueList.type = C
Value1.key = 0
Value1.value = *language*customer.extName4.keyValueList.value1
Value2.key = 1

```

```
Value2.value = *language*customer.extName4.keyValueList.value2  
Value3.key = 2  
Value3.value = *language*customer.extName4.keyValueList.value3
```

Chapter 9

Building a customer client

All Thin Client deliveries are customer dependent. There is no core Thin Client. To provide a deliverable package for the customer at least one file (build.properties) must be branched and modified for the customer. The package will be built and distributed to the projects folder for the respective SignPlus release.

For each SignPlus Release there is a branch for the entire core release.

Example

```
TClient#core#REL-4-1
```

```
TClient#core#REL-4-2
```

```
TClient#core#REL-4-3
```

and so on.

The core branches are required when building a customer release. First the base release will be checked out from core and the updated to contain the branched files for the actual customer branch. The customer branch contains only the files which are modified for the customer.

Branching and customizations

When building a customer specific client the files customized for the customer will be branched from the current release. This requires that the build.properties at least be branched and modified. The build.properties includes the build specifications for the current customer and is always required.

Should any customer specific modifications be required these files will also branched and modified as required. Branch tags are always created as follows:

```
TClient#'client short name'#REL-'major'-'minor'
```

where:

'client short name' is the client short name used for cvs

'major' is the major SignPlus version

'minor' is the minor SignPlus version

Tagging

When tagging the Thin Client all modules are tagged. The entire release will be checked out during the build process.

```
RST#TClient#'client short name'#REL-'major'-'minor'-'release'
```

where:

'client short name' is the client short name used for cvs

'major' is the major SignPlus version

'minor' is the minor SignPlus version

'release' is the current release for the thin client (0 -> unlimited)

Building

To builds the Thin Client either the complete project is required in an Eclipse environment or the Build Server can be used.

Build Server

For all requirements see BThinClient.txt on the build server. This explains what input and prerequisites are required.

Input required:

```
BThinClient.cmd TAG [build_type]
```

Example

```
BThinClient.cmd TClient#key_bank#REL-4-2-4
```

Parameter	Description
TAG	<p>The cvs tag (without RST#): always required</p> <p>Example</p> <p>To build tag: Input:</p> <pre>RST#TClient#key_bank#REL-4-2-0 TClient#key_bank#REL-4-2-0 RST#TClient#fcb_tt#REL-4-2-0 TClient#fcb_tt#REL-4-2-0</pre> <p>Note When 'RST#' is located in the TAG input an error will occur. The build will fail.</p>

Parameter	Description
build_type	<p>The build_type is required for core components, optional for customer releases. Supported build_type values are:</p> <ul style="list-style-type: none">custom - Non core customer release (default, not required for customer release)fraudone - FraudOne build, can be used only when the TAG contains #core#demobank - DemoBank build, can be used only when the TAG contains #core#signinfo - SignInfo build, can be used only when the TAG contains #core# <p>Note When 'custom' is used and #core# is located in the TAG an error will occur. The build will fail.</p> <p>When no build_type is used and #core# is located in the TAG an error will occur. The build will fail.</p> <p>When no build_type is used and #core# is NOT located in the TAG the default 'custom' will occur.</p> <p>When 'fraudone' is used and #core# is NOT located in the TAG an error will occur. The build will fail.</p> <p>When 'demobank' is used and #core# is NOT located in the TAG an error will occur. The build will fail.</p> <p>When 'signinfo' is used and #core# is NOT located in the TAG an error will occur. The build will fail.</p>