Kofax FraudOne
Thin Client Customization Guide

Version: 4.6.0

Date: 2022-10-17

KOFAX

Table of Contents

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/4.6.0-e4jy6kf7pr/FO.htm

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

Release notes
- *Kofax FraudOne Release Notes*

Technical specifications
- *Kofax FraudOne Technical Specifications*

Guides
- *Kofax FraudOne Administrator's Guide*
- *Kofax FraudOne Archive Interface Server*
- *Kofax FraudOne ASV Blackbox*
- *Kofax FraudOne Common API Specifications for GIA Engines*
- *Kofax FraudOne Data Warehouse Installation and Operation Guide*
- *Kofax FraudOne Extended Reporting Features and Statistics*
- *Kofax FraudOne Feature Codes*
- *Kofax FraudOne Global Fraud Signature Web Service Developer's Guide*
- *Kofax FraudOne Installation and Migration Guide*
- *Kofax FraudOne Java Client Customization Guide*
- *Kofax FraudOne Java Client Customization Layer*
- *Kofax FraudOne Report Component Installation Guide*
- *Kofax FraudOne Service Program Configuration*
- *Kofax FraudOne Service Program Interfaces*
- *Kofax FraudOne SignCheck Result Codes*

- *Kofax FraudOne Standard Reporting Features and Statistics*
- *Kofax FraudOne The Book on CRS*
- *Kofax FraudOne Variant Cleanup Utility*

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

Training

Kofax offers both classroom and online training to help you make the most of your product. To learn more about training courses and schedules, visit the Kofax Education Portal on the Kofax website.

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:

1. Go to the Kofax website home page and select **Support**.

2. When the Support page appears, select **Customer Support** > **Knowledge Base**.

> ⓘ 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.

From the Knowledge Base home page, you can:
- Access the Kofax Community (for all customers).

  Click the **Community** link at the top of the page.
- Access the Kofax Customer Portal (for eligible customers).

Click the **Support** link at the top of the page. When the Customer & Partner Portals Overview appears, click **Log in to the Customer Portal**.

- Access the Kofax Partner Portal (for eligible partners).

  Click the **Support** link at the top of the page. When the Customer & Partner Portals Overview appears, click **Log in to the Partner Portal**.

- Access Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.

  Go 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 with 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)
ARestictions.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)
bno.properties	Common	WEB-INF/lib/spclient.jar (de/softpro/signplus/resources)

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

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.

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 placed 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.

General and view settings

This section describes 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.

DisableRightMouseClick

Example

```
DisableRightMouseClick = 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 FraudOne 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.

displayLogo.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 FraudOne 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 FraudOne 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. The 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 valued 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

Usage	Right	Binary value	Decimal
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
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	65563

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`).

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

ⓘ (*) For MS-SQL 2005 and higher. 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

Allows the font to be used for the Watermark to be specified. The fonts installed on your system (server) where the Thin Client is installed is allowed. If the font is not located then Arial will be used as a default.

WatermarkFont

Example

```
WatermarkFont = Lucida
```

Font size for the watermark

Allows the font size to be used for the Watermark to be specified.

WatermarkFontSize

Example

```
WatermarkFontSize = 28
```

Default: 20

Font bold for the watermark

Insert the watermark using 'Bold' if specified.

WatermarkFontBold

Example

```
WatermarkFontBold = true
```

Default: false

Font italic for the watermark

Insert the watermark using 'Italic' if specified.

WatermarkFontItalic

Example

```
WatermarkFontItalic = true
```

Default: false

Opacity for the watermark

Insert the watermark using the desired level of opacity. Values allowed are from 20 to 100. Increasing the setting increases the opacity of the watermark.

WatermarkOpacity

Example

```
WatermarkOpacity = 60
```

Default: 50

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

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 preferred, 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 preferred.

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 preferred.

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 preferred 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

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

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

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 FraudOne server and should be used there where possible. This may be deprecated later.

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-ImageId.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

EasyExtensions

EasyExtensions is a configuration capability which is performed solely within the custom.zip. The extension data is displayed following the FraudOne 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 *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:
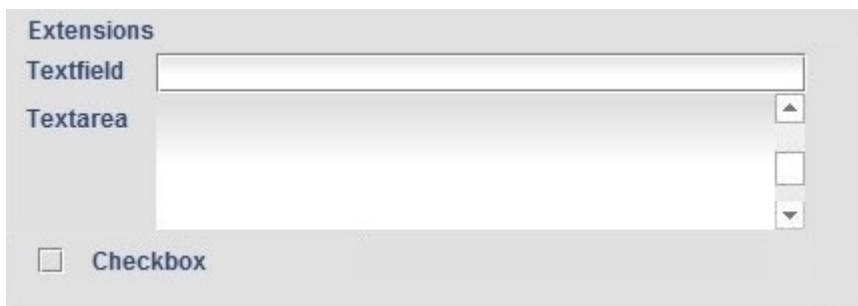- SignBase.properties (SignBase.properties.tclient)
- lists.properties (lists.properties.tclient)
- language.properties (language.properties.tclient)
- The language.properties for customer specific language files (ie. language_de.properties) are also processed.

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.

ⓘ 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

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

Extension Type	Notes
6 (Date)	[Not supported for checkbox]
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 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.

Account Details

Signatories Details

Signatory		Instruction
Last Name	Any	
First Name	Body	
Middle Name		
Role	owner	
Title		
Personal-Id		
Position		
Birth Date		None
Member of Group(s)	Group(s)A	
Extensions Header		
Ext.5	S1	
Ext.6	S1 Loooong	
☑ Ext.7		
Ext.8	Signo text for value 1	Valid until

Image		
Captured from		
Capture date		
Captured as		
Date Signed	10/9/15	
type	signature	**X**
Signature style	latin	
☐ good quality		
☐ not for ASV		

Mandates

	Right	Amount	Currency	From	Until	Who	With whom
	Single	1,000	USD	10/7/15	10/10/15	(A) GrA	

Rules Details

Details for rule 1. Single	
Right	Single
Amount	No Limit
Currency	
Who	Noo, Body
Extensions Header	
Ext.9	RuleText
Ext.10	Long RuleText
☑ Ext.11	
Ext.12	Rule text for value 3

Check Stock

Image type	normal
Document Id	1
APIA enabled	☑
Scan date	10/9/15
Valid from	12/8/15
Extensions Header	
Ext.17	SI 1
Ext.18	Si1 Looooong
Ext.19	☑
Ext.20	Stock Image text for value 1
Instruction	No instruction
Instruction Code	0
Valid till	
UID (instruction)	

Account Image

Account 71034
  1 – test (71034)
  2 (71034)
  3 (71034)

Image Nam

Image-Id	12
Text 1	test
Text 2	
Scanned	10/12/15
Last Changed	10/21/15
Processed	☑
Remarks	None
Extensions Header	
Ext.21	AA1
Ext.22	AccountAttachment
Ext.23	☑
Ext.24	Account Image text for value 1

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
SSBAM_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
```