

SignAnyWhere Viewer - Customization

- Customization of localizations
- Customization of the SAW Viewer Layout (custom.css inside Customization.zip)
 - Example: Hide the Mouse option for biometric signatures
 - Example: Hide the SignOnOhone option for biometric signatures
- Example: Change Default Color and Logo
- Viewer Preferences
- Document Policies
- Configurations
- SignAnyWhere Viewer – Opening Link
- SignAnyWhere Viewer Client Finish Action

This advanced feature is for customizing the **eSignAnyWhere Viewer** for the signers. You can change the colors and logo to align them to your CI and set eSAW Viewer behavior. Therefore you have to upload a ZIP-Archive with the new design settings. You can download a design template and change it. Moreover you are able to set a redirect URL when a document is finished.

You can find the the configuration for the customization of eSignAnyWhere in your organization settings in the section "Design of the document viewer for recipients"

The next screenshot shows these configurations:

Design of the document viewer for recipients

The screenshot shows a configuration interface with the following elements:

- Default redirect URL for finished documents:** A text input field containing the placeholder text "Default redirect URL for finished documents".
- Viewer Design:** A section containing four buttons:
 - UPLOAD DESIGN
 - RESET TO DEFAULT
 - DOWNLOAD CURRENT DESIGN (with a download icon)
 - DOWNLOAD DESIGN TEMPLATE (with a download icon)
- Biometric encryption key:** A text input field that is currently blurred.

If the feature is available for your organization, you can:

- set a redirect URL for finished documents
- upload a design
- reset to default design
- download a design template
- download current design

The *Customisation.zip* file contains:

- *variables.xml*: contains the style configuration of eSignAnyWhere. The comments in the file will help you to modify it.
- *global_variables.xml*: contains settings for the eSAW Viewer
 - **Note:** In the *global_variables.xml* you can find the variable *ContentSecurityPolicy*. This variable allows you to integrate the viewer in an IFrame. Just enter the page URL in the "value" area on which you want to integrate the viewer. In addition to the possibility of the customization zip there are additional configurations for further scenarios available via API; and via *_global.xml* for on-premise. These are only required in exceptional cases or for special integration scenarios. Example: *WhitelistedPostMessageClients*
- *files/*

In the files you can find the following:

- *logo_topbar*
- *logo_loading*
- *Localizations.template.custom*
 - Here you can customize localizations for a specific eSAW organization
 - You can create your own *Localizations.*.custom.json* files, modify the elements which should be customized and add the file to the customization package
- *custom.css*
 - Here you can customize the layout for a specific eSAW organization
 - You can adapt the layout of the SAW Viewer

If you have modified the files just put them in an archive and upload via UI. You can use svg instead of png, gif or jpg.

Customization of localizations v 21.16



Following feature flag is necessary for this feature: UseCustomizationId

Please note the following: This is a non-default feature of eSignAnyWhere. If you are interested in this optional feature please [contact us](#).

In detail the following options are possible:

- Customize localizations for a specific eSAW organization (only possible in combination with eSAW and Customization Service):
 - The WSC customization package contains a Localizations.template.custom.json containing all keys & english values of the XLF but in JSON format.
 - You can create your own Localizations.*.custom.json files (e.g. [Localizations.de.custom.json](#)), modify the elements which should be customized and add the file to the customization package.
 - For better maintenance all items which have not been changed should be removed from the JSON file.

Note: Not all sections of the XLF file are allowed to be configured that way (e.g. to prevent possible problems with changing audit trail texts)

- Customize localizations for the whole SSP instance (or in case Customization Service is not used):
 - Grab the Localizations.template.custom.json from the default customization service package (see Release Notes)
 - Modify the file like described above (content, filename)
 - Place the file in the i18n folder next to the XLF files
- Combination of these two possibilities
 - In case of a 'combined' customization (JSON files available in Customization Service and i18n folder) the two JSON files will be merged. In case items are included in both files, the items from the Customization Service will override the ones from the i18n folder.

Please note: It is not recommended anymore to change the XLF files (SignAnyWhere Viewer) directly!

Customization of the SAW Viewer Layout (custom.css inside Customization.zip)

The following properties are allowed to change:

```
"display": {},
"visibility": {},
"width": {},
"height": {},
"border-radius": {},
"margin": {},
"margin-top": {},
"margin-bottom": {},
"margin-left": {},
"margin-right": {},
"padding": {},
"padding-top": {},
"padding-bottom": {},
"padding-left": {},
"padding-right": {},
"color": {},
"background-color": {},
"background-repeat": {},
"background-position": {},
"border-color": {}
```

Please also see the next sample:

```
#topBar {
    background-color: red
}

#leftBarView{
    visibility: hidden
}
```

Example: Hide the Mouse option for biometric signatures

Please note that both elements were a "div" element before version 21.16. With version 21.16 and the versions afterwards both elements were changed to a "button" element.

Before the version 21.16 please see the following configuration for hiding mouse and SignOnPhone

```
div[title='Mouse'] {  
    visibility: hidden;  
}
```

With 21.16 and the versions afterwards see the following configuration for hiding mouse and SignOnPhone

```
button[data-id='Mouse'] {  
    visibility: hidden;  
}
```

Example: Hide the SignOnPhone option for biometric signatures

This example removes both the "SignOnPhone" option from the device selection dialog, but also the QR code and App links from the SignOnPhone dialog shown when no SIGNificant Device Driver was found.

Note: the dialog itself, its buttons and the descriptive text of the dialog remain visible. Other than when using the instance-wide custom.css, you cannot replace the text shown on top. Only in an instance-wide custom.css you can replace this text.

Before the version 21.16 please see the following configuration for hiding mouse and SignOnPhone

```
div[title='SignOnPhone'] {  
    visibility: hidden;  
}  
  
#sopProtocollLink{  
visibility:hidden;  
}  
  
.inSigningProgress{  
visibility:hidden;  
}  
  
#qrcodeImage > canvas{  
    visibility:hidden;  
    display: none;  
}
```

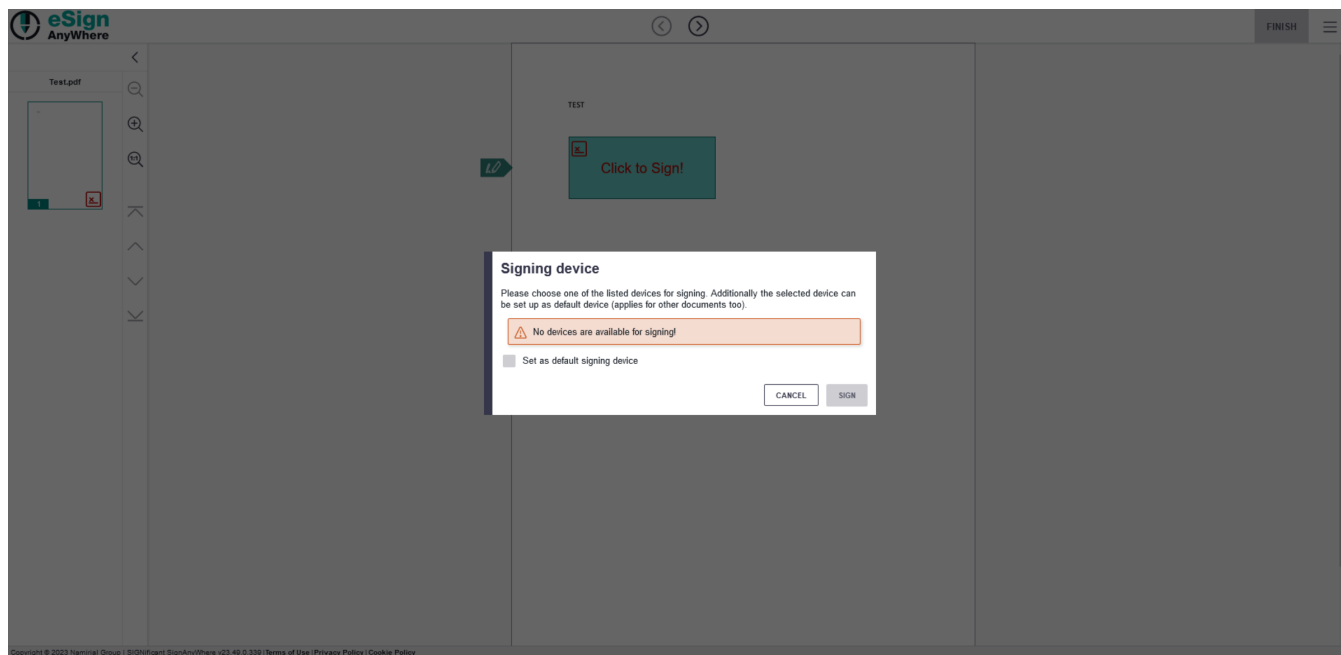
With 21.16 and the versions afterwards see the following configuration for hiding mouse and SignOnPhone

```
button[data-id='SignOnPhone'] {  
    visibility: hidden;  
}  
  
#sopProtocollLink{  
visibility:hidden;  
}  
  
.inSigningProgress{  
visibility:hidden;  
}  
  
#qrcodeImage > canvas{  
    visibility:hidden;  
    display: none;  
}
```

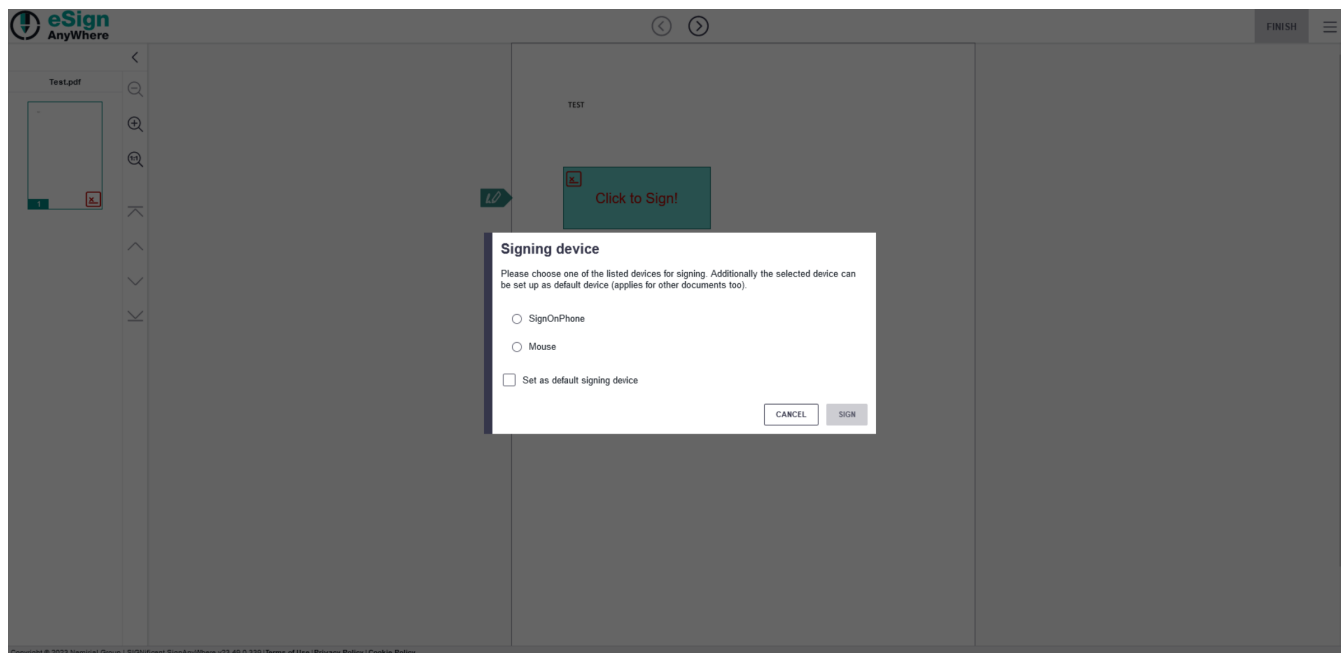
 Please note that since version 22.28 it is possible to disable and enable SignOnPhone and Mouse via the customization.zip in the global_variables.xml with the following two variables:

```
<variable name="DisableSignOnPhone"/>
<variable name="DisableMouseForBiometricSignature"/>
```

If you removed both Mouse and SignOnPhone and no other device for signing was found, the device selection looks like:



Without those configuration (Mouse and SignOnPhone enabled) the following settings will be visible:



For an instance-wide custom.css (not within Customization.zip) you could change the descriptive text above/instead the QR code image with following CSS definition:

```
#sopMainPanel:after{
content: 'Attention: The sender defined to record a biometric (handwritten) signature. This requires to install the Device Driver.';
}
```

Example: Change Default Color and Logo

Open variables.xml and look for the following lines of code to change the colors of for example the page navigation background.

Replace the following line:

```
<variable name="primaryColor" value="#{[color:#00A69D]}#" comment="
    default: #00A69D Primary color of the SAW Viewer affecting several major elements and referenced in
different shades like 'primaryLightestColor' or 'primaryLighterColor' (see 'ADVANCED' section)
" category="Basic" />
```

With this one:

```
<variable name="primaryColor" value="#{[color:#000088]}#" comment="
    default: #000088 Primary color of the SAW Viewer affecting several major elements and referenced in
different shades like 'primaryLightestColor' or 'primaryLighterColor' (see 'ADVANCED' section)
" category="Basic" />
```

Replace the following line:

```
<variable name="secondaryColor" value="#{[color:#35364A]}#" comment="
    default: #35364A Secondary color of the SAW Viewer affecting most icons, some buttons and used as
counterpart of 'primaryColor'
" category="Basic" />
```

With this one:

```
<variable name="secondaryColor" value="#{[color:#000088]}#" comment="
    default: #000088 Secondary color of the SAW Viewer affecting most icons, some buttons and used as
counterpart of 'primaryColor'
" category="Basic" />
```

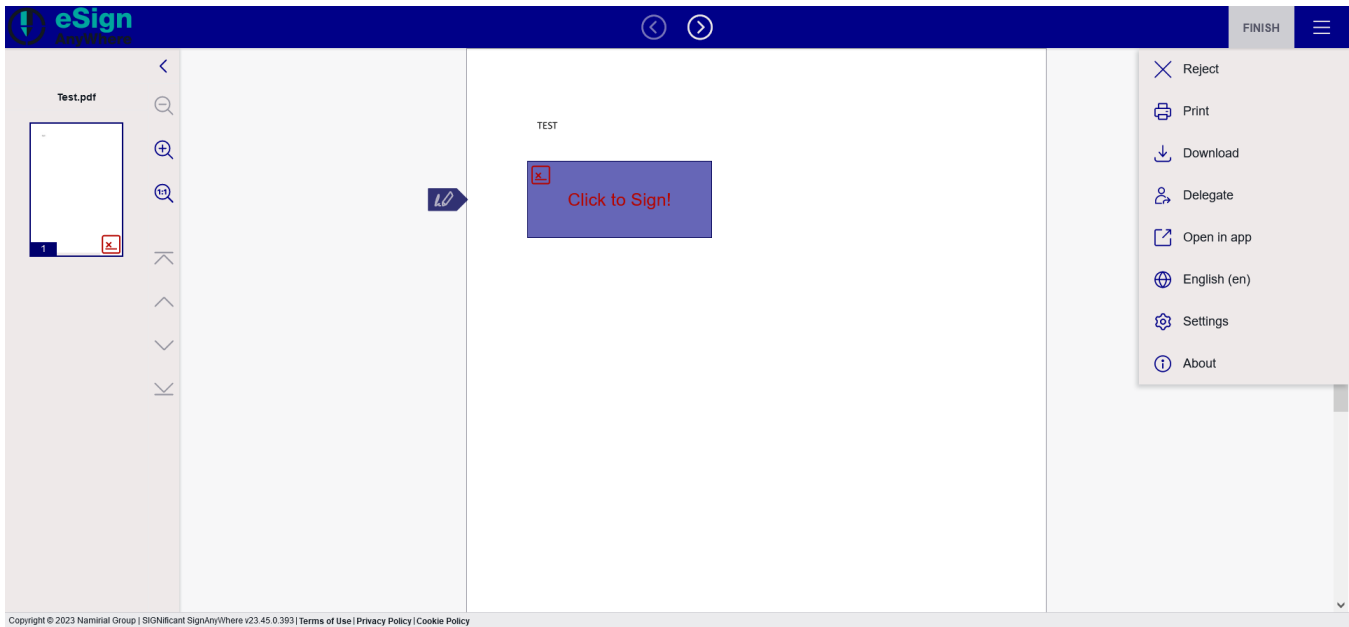
Replace the following lines:

```
<variable name="guidanceButtonNextColor" value="#{[colorvar:secondaryColor]}#" comment="" category="Expert" />
<variable name="guidanceButtonPreviousColor" value="#{[colorvar:secondaryColor]}#" comment="" category="Expert" />
<variable name="menuOverlayButtonColor" value="#{[colorvar:secondaryColor]}#" comment="" category="Expert" />
```

With this one:

```
<variable name="guidanceButtonNextColor" value="#{[colorvar:whiteColor]}#" comment="" category="Expert" />
<variable name="guidanceButtonPreviousColor" value="#{[colorvar:whiteColor]}#" comment="" category="Expert" />
<variable name="menuOverlayButtonColor" value="#{[colorvar:whiteColor]}#" comment="" category="Expert" />
```

After this configurations the viewer should look like the following:



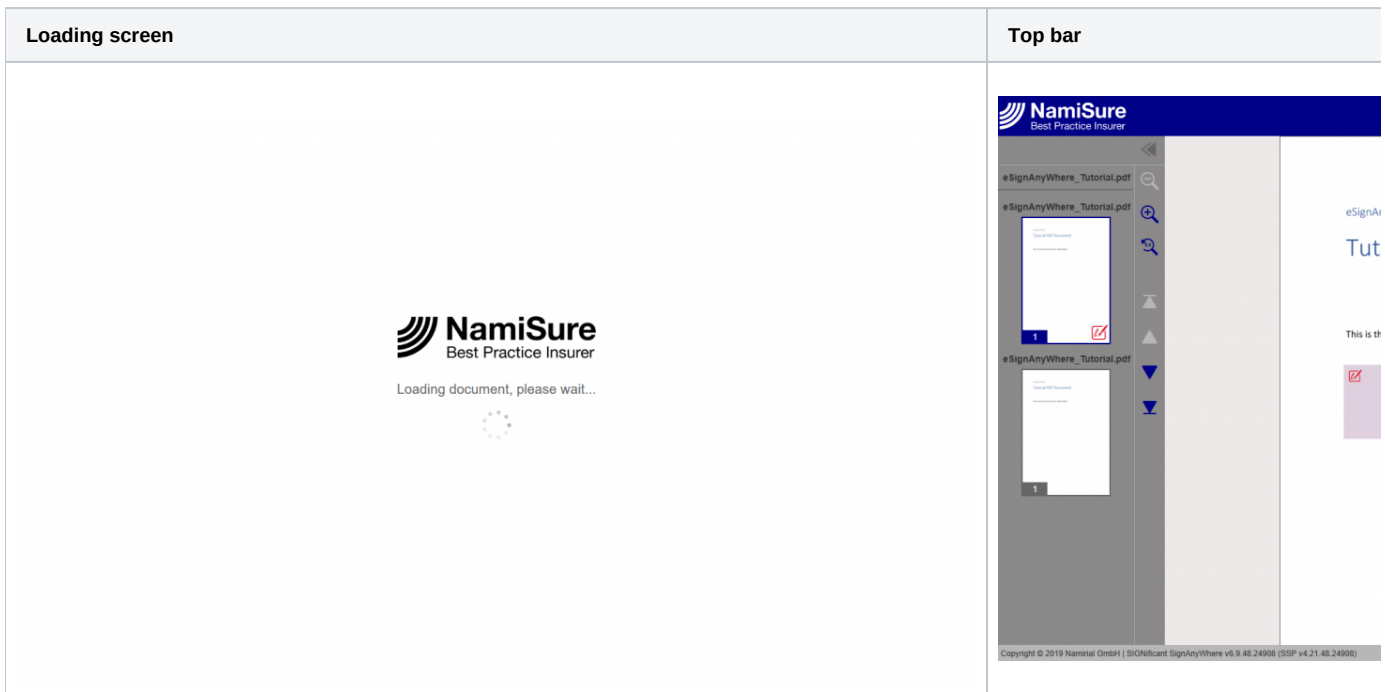
For changing the Logo you have to put your Logo in the subfolder `/files`. Then you have to change the `variables.xml` to set the new files (e.g. my-logo-large.png and my-logo-small.png).

```
<variable name="logoCollapsedUrl" value="#{image:my-logo-small.png}}#" ... />
<variable name="logoExpandedUrl" value="#{image:my-logo-large.png}}#" ... />
```

Save the files and pack it in a zip-file (e.g. "my-style.zip"), with same structure as the downloaded *Customization.zip*, and upload it as new design in your organization settings.

There are to places in the XML where you can define the logo:

- For the loading screen
- and for the top bar



 It may take some minutes until the new design is applied to your eSignAnyWhere Viewer.

Viewer Preferences

 Note that changes of the viewer preferences will be effective within some minutes after uploading the changed Customization.zip file. The changes are applied also on existing envelopes. Note that changes may have legal effect even on older envelopes (e.g. enabling/disabling the SkipFinishConfirmDialog, FinishWorkstepOnOpen, AcceptAgreementDisabledUntilRequiredActionsDone but also others); consider the impact of a change carefully.

Name	Description	Values	Note
MessageForwardingDialogBehavior	Either 'Show' or 'Allow'; If 'Show' is set and forwarding JS PostMessages to e.g. a parent iFrame when setting PostmessageClient query parameter the user will get asked if he/she wants to allow this. When 'Allow' is set, the user will not get asked and sending messages to the iFrame automatically gets allowed.	Possible values: 0, 1 (default)	
ShowPageNavigationBar	Allows to disable the page navigation bar on the left side of the viewer	Possible values: 0, 1 (default)	Deprecated => Use ShowPageNavigationButtons instead
ShowThumbnails	If enabled, the thumbnail display will be visible, otherwise false	Possible values: 0, 1 (default)	
SkipFinishConfirmDialog	If enabled, finish action will be executed immediately without confirm dialog	Possible values: 0 (default), 1	
SkipDocumentDialog	If enabled, the download document dialog after finishing the workstep will be skipped	Possible values: 0 (default), 1	
ShowImagesInFullWidth	If enabled, available width will be used for images which might stretch them	Possible values: 0 (default), 1	
DisableGeolocation	If enabled, Geolocation won't be used for webservice calls	Possible values: 0 (default), 1	
ShowDocumentDownloadDialogAfterAutomaticFinish	If enabled, the "Download documents dialog" after automatically finishing the workstep will be shown (e.g. acknowledge receipt scenario)	Possible values: 0 (default), 1	
AttachmentsMaxFileSize	Allows to configure the maximum file size for attachments in kBytes	Possible values: Any integer value, (default: 2048)	Not possible via customization
SkipPreviewImageOnDisposableCertificate	Allows to disable the preview image of the signature when using the Namirial Disposable certificate functionality	Possible values: 0, 1 (default)	
LoadCustomJs	If enabled, a Custom.js file will be loaded when the page is opened allowing to configure customized data	Possible values: 0 (default), 1	Not possible via customization
AllowCustomButtons	If enabled, custom buttons can be added via xyzmo.Customization plugin	Possible values: 0 (default), 1	Not possible via customization
GuidingBehavior	Defines how the viewer should guide through the tasks	Possible values: GuideOnlyRequiredTasks, GuideRequiredAndOptionalTasks (default), GuideDisabled	
FormFieldsGuidingBehavior	Defining how the viewer should guide through form fields	Possible values: AllowSubmitAlways (default), AllowSubmitOnlyAfterAllRequiredFieldsAreDone	
ShowVersionNumber	Defining if the viewer should show the version number	Possible values: 0, 1 (default)	Deprecated => Use ShowStatusBar instead
EnableWarningPopupOnLeave	Defining if the viewer should display a warning popup when the user leaves the page	Possible values: 0, 1 (default)	
WarningPopupDisplayAfter	Defining when the warning popup should be displayed	Possible values: FillOrSignField (default), Authentication, Always, Agreement	

FinishWorkstepOnOpen	If set to 1 the workstep is automatically finished when opened in the viewer, if finishing is possible	Possible values: 0 (default), 1	Not possible via customization
MultipleSignatureTypesAndBatchSigningSettings / IsUseBatchSigningCheckedByDefault	If enabled, the "Use batch signing" checkbox inside the dialog when signing a signature field is checked by default	Possible values: 0, 1 (default)	
MultipleSignatureTypesAndBatchSigningSettings / IsRememberSignatureTypeCheckedByDefault	If enabled, the "Remember signature type" checkbox inside the dialog when signing a signature field is checked by default	Possible values: 0 (default), 1	
MultipleSignatureTypesAndBatchSigningSettings / IsRememberBatchSigningDecisionCheckedByDefault	If enabled, the "remember batch signing decision" checkbox inside the dialog when signing a signature field is checked by default	Possible values: 0 (default), 1	
MultipleSignatureTypesAndBatchSigningSettings / SkipMultipleSignatureTypesAndBatchSigningDialogIfBatchSigningPossible	If enabled, and the IsUseBatchSigningCheckedByDefault is also enabled, the dialog to confirm the batch signing (and optionally choosing the signature type) will be skipped if either the signature type to use has been remembered earlier or a preferred type is set	Possible values: 0 (default), 1	
AutoFinishAfterRequiredTasksDone	If enabled, finish action will be automatically triggered after last required task is done	Possible values: 0 (default), 1	
GuidingBehaviorOnFinishedTask	Defines the guiding behavior after a task has been completed (e.g. signature field has been signed). This flag is not taken into account if the guiding is disabled	Possible values: NoMove, MoveToNext (default), MoveToNextAndActivate	
SkipThankYouDialog	If enabled, hints that the document can now be finished after all required tasks have been done, will be shown	Possible values: 0 (default), 1	Deprecated => Use ShowFinishPossibleHint instead
DisplayRejectButtonInLeftBar (replaced by DisplayRejectButtonInTopBar in SSP 4.15.54 or higher)	If enabled, the 'reject' button will be displayed in the left bar below the 'finish' button as well	Possible values: 0 (default), 1	
NativeAppsUrlScheme	Defines the url scheme for the native SAW Viewer applications (needed for customized SAW Viewer apps)	Possible values: Any custom URL scheme (default: esignanywhere)	Not possible via customization
DocumentViewingMode	Defines whether the envelope is displayed in its entirety or if only a specific document is shown at a time	Possible values: EndlessPaperAllDocuments (default), EndlessPaperPerDocument	
ThumbnailMode	Defines the appearance of the thumbnail view: all pages of the entire document are shown, or only the first page of each document within an envelope	Possible values: ShowAllPages (default), ShowFirstPageOnly	
ShowTopBar	Defines if the top bar should be visible or not. If disabled, also the 'GuidingBehavior' and the 'DisplayRejectButtonInTopBar' will be deactivated.	Possible values: 0, 1 (default)	
DisplayRejectButtonInTopBar (replaces DisplayRejectButtonInLeftBar from 4.15.54)	If enabled, the 'reject' button will be displayed in the top bar next to the 'finish' button as well.	Possible values: 0 (default), 1	
PhoneNumberInputSettings / DefaultCountry	If 'auto' is defined, the first one of the list is preselected or use ISO 3166-1 alpha-2 code (f.e. 'it')	Possible values: auto (default), ISO 3166-1 alpha-2 code (f.e. 'it')	
PhoneNumberInputSettings / PreferredCountryList	If 'auto' is defined, the first one of the list is preselected or use ISO 3166-1 alpha-2 code (f.e. 'it')	Possible values: it,ro,at,de (default), ISO 3166-1 alpha-2 code (f.e. 'it')	
TriggerBankIDDesktopApp	Defines if the BankID desktop app should be triggered from the browser when authenticating via BankID.	Possible values: 0 (default), 1	Not possible via customization
ShowPageGap	Defines whether or not to show a small gap between pages containing some bits of information (like corresponding document and page x/y) in Desktop/Tablet UI	Possible values: 0, 1 (default)	
AutoStartGuiding	Defines if the guiding (if disabled at all) should be started directly upon start.	Possible values: 0 (default), 1	
ShowFinishPossibleHint	If enabled, hints that the document can now be finished after all required tasks have been done, will be shown	Possible values: 0, 1 (default)	
ShowStartGuidingHint	If enabled, a hint to the next button is shown, where the guide can be started	Possible values: 0, 1 (default)	
ShowStatusBar	Defines if the status bar at the bottom should be visible or not. If disabled, product and version number and biometric signature device selection window is not available	Possible values: 0, 1 (default)	

ShowPageNavigationButtons	allows to disable the page navigation buttons on the left side of the viewer	Possible values: 0, 1 (default)	
ShowZoomButtons	allows to disable the zoom buttons on the left side of the viewer	Possible values: 0, 1 (default)	
ShowNoGeolocationWarning	Defines if a warning should be shown in case the browser does not allow to retrieve the geolocation.	Possible values: 0 (default), 1	Not possible via customization
SkipRejectConfirmDialog	If enabled, reject action will be executed immediately without confirm dialog	Possible values: 0 (default), 1	
DisableSignOnPhone	Defines if SignOnPhone (if configured appropriately) should be offered as an option to sign biometrically.	Possible values: 0 (default), 1	
DisableMouseForBiometricSignature	Defines if 'Mouse' should be offered as an option to sign biometrically upon using SIGNificant Device Driver.	Possible values: 1 (default), 0	
SignatureComplexityChecks/Draw2Sign/MinimumPackets	Minimum count/amount of packets for a signature. Default is empty value or 0 => All signatures are accepted in terms of length	Default: 0 (disabled)	
SignatureComplexityChecks/Draw2Sign/MinimumTimeInMs	Minimum time in milliseconds for a signature. Default is empty value or 0 => All signatures are accepted in terms of duration	Default: 0 (disabled)	

Document Policies

These policies, also called "Global Policies" sometimes, define which of the features of a client are allowed on a signer activity - in case the client application supports this feature. Please note that the SignAnyWhere Viewer does not support all of these.

Name	Description	Values	SAW-Viewer
AllowSaveDocument	Is the client allowed to save the workstep document	Possible values: 0 (default), 1	✓
AllowSaveAuditTrail	Is the client allowed to save the audittrail document	Possible values: 0 (default), 1	✓
AllowRotatingPages	Is the client allowed to rotate the pages of the document	Possible values: 0, 1 (default)	✗
AllowAppendFileToWorkstep	Is the client allowed to append a file to the workstep	Possible values: 0 (default), 1	✗
AllowAppendTasksToWorkstep	Is the client allowed to add append tasks to the workstep	Possible values: 0 (default), 1	✗
AllowEmailDocument	Is the client allowed to email the workstep document	Possible values: 0 (default), 1	✗
AllowPrintDocument	Is the client allowed to print the workstep document	Possible values: 0 (default), 1	✓
AllowFinishWorkstep	Is the client allowed to finish the workstep	Possible values: 0, 1 (default)	✓
AllowRejectWorkstep	Is the client allowed to reject the workstep	Possible values: 0 (default), 1	✓
AllowRejectWorkstepDelegation	Is the client allowed to reject the workstep with reject type 'Delegation'	Possible values: 0, 1 (default)	✓
AllowUndoLastAction	Is the client allowed to undo actions	Possible values: 0 (default), 1	✓
AllowColorizePdfForms	Is the client allowed to colorize pdf forms	Possible values: 0 (default), 1	✗
AllowReloadOfFinishedWorkstep	Is the client allowed to view/download the document after finishing	Possible values: 0, 1 (default)	✓
AllowAdhocPdfAttachments	Is the client allowed to add adhoc attachments	Possible values: 0 (default), 1	✓
AllowAdhocSignatures	Is the client allowed to add signatures which are not defined in the tasks	Possible values: 0 (default), 1	✗
AllowAdhocStampings	Is the client allowed to add stampings which are not defined in the tasks	Possible values: 0 (default), 1	✗
AllowAdhocFreeHandAnnotations	Is the client allowed to add adhoc freehand annotations	Possible values: 0 (default), 1	✗
AllowAdhocTypewriterAnnotations	Is the client allowed to add adhoc typewriter annotations	Possible values: 0 (default), 1	✗
AllowAdhocPictureAnnotations	Is the client allowed to add adhoc picture annotations	Possible values: 0, 1 (default)	✗
AllowAdhocPdfPageAppending	Is the client allowed to append adhoc documents	Possible values: 0 (default), 1	✗

Configurations

Configurations for API integrations, where you directly configure the workstep per recipient.

WorkstepIdStampConfiguration

In the configuration of the workstep you can define the stamp imprint:

```
<workstepIdStampConfiguration>
  <referenceCorner>Upper_Left</referenceCorner>
  <color>White</color>
  <location width="340" height="15" marginLeft="3" marginTop="3" />
</workstepIdStampConfiguration>
```

NotificationConfiguration

Also the notification configuration is possible via the workstep configuration.

```
<NotificationConfiguration>
  <NotificationEventsNotToSend>
    <NotificationEvent>ActionCalledEvent</NotificationEvent>
    <NotificationEvent>WorkstepCreated</NotificationEvent>
    <NotificationEvent>SignWorkstepDocument</NotificationEvent>
  </NotificationEventsNotToSend>
  <NotificationUrl>http://localhost:52094/ReceiverSample.aspx?
WorkstepId=##WorkstepId##&EventType=##EventType##&Source=##Source##&Time=##Time##&Description=##Description<
/NotificationUrl>
</NotificationConfiguration>
```

StampImprintConfiguration

You also can define the stamp imprint for signature fields configuration.

```
<AllowedSignatureTypes>
  <sigType id="..." type="Picture">
    <allowedCapturingMethods>Click2Sign</allowedCapturingMethods>
    <StampImprintConfiguration>
      <DisplayExtraInformation>1</DisplayExtraInformation>
      <DisplayEmail>1</DisplayEmail>
      <DisplayIp>0</DisplayIp>
      <DisplayName>1</DisplayName>
      <DisplaySignatureDate>1</DisplaySignatureDate>
      <FontFamily>Times New Roman</FontFamily>
      <FontSize>11</FontSize>
    </StampImprintConfiguration>
  </sigType>
</AllowedSignatureTypes>
</sig>
```

SignAnyWhere Viewer – Opening Link

Opening a workstep with the SAW Viewer is quite easy. Just reference to the SAW Viewer endpoint and use a workstepid parameter.

<http|https>://<server>/<virtualDirectory>/SignAnywhere.html?WorkstepId=<WorkstepId>

Parameters

Parameter	Values	Description
setLng	(de en it ...)	defines the UI language, if not defined the browser language is used if available, otherwise fallback to English
isRemoteDesktop	(0 1)	defines if Device Driver should establish a connection to DeviceDriver Client Components. Required in Citrix environments when not configured in _global.xml
ignoreDeviceDriver	(0 1)	defines that no device driver should be loaded. Was implemented for situations where SignOnPhone should be used.

SignAnyWhere Viewer Client Finish Action

You can configure a client finish action to be redirected to an predefined URL for the signer.

```
<FinishAction>
  ...
  <ClientAction CallClientActionOnlyAfterSuccessfulSync="1" clientName="SIGNificant SignAnywhere"
>http://myurl.com</ClientAction>
  ...
</FinishAction>
```