AusweisApp
Extended documentation for administrators and developers

2.5.1

Governikus GmbH & Co. KG

Table of contents

1 Installation

1.1 Windows

Start the installer of AusweisApp using the command line to configure the installation process and preset system-wide default settings. The return value of msiexec indicates the result of the installation[1]. In addition to the usual arguments[2], the following command contains all supported arguments, which are explained below.

```
msiexec /i AusweisApp-X.YY.Z.msi /quiet INSTALLDIR="C:\AusweisApp" SYSTEMSETTING␣
→S=false DESKTOPSHORTCUT=false PROXYSERVICE=false AUTOSTART=false TRAYICON=true A␣
→UTOHIDE=false REMINDTOCLOSE=false ASSISTANT=false TRANSPORTPINREMINDER=false CUS␣
→TOMPROXYTYPE="HTTP" CUSTOMPROXYHOST="proxy.example.org" CUSTOMPROXYPORT=1337 UPD␣
→ATECHECK=false SHUFFLESCREENKEYBOARD=true SECURESCREENKEYBOARD=true ENABLECANALL␣
→OWED=true SKIPRIGHTSONCANALLOWED=true LAUNCH=true
```

INSTALLDIR States the installation directory. If not specified, the folder "C:\Program Files\AusweisApp" is used.

SYSTEMSETTINGS Concerns the settings of firewall rules of the Windows Firewall. When not specifying the argument, firewall rules are created (true). By indicating SYSTEMSETTINGS=false, no firewall rules are created.

DESKTOPSHORTCUT By specifying DESKTOPSHORTCUT=false, no desktop shortcut is created. Without specifying the argument, the desktop shortcut is created for all users (true).

PROXYSERVICE The proxy service is required to enable the parallel operation of several entities of AusweisApp. The proxy service monitors port 24727 (defined in BSI TR-03124-1) and forwards requests to the local AusweisApp instances. The Discovery messages (amendment to BSI TR-03112-6 - IFD Service - Chapter 3) are not forwarded, so that SaC devices cannot be recognized or used in this operating mode. Not specified, the proxy service will be installed automatically if Terminal Services is installed and the system is running in application server mode.

AUTOSTART By setting AUTOSTART=true, an autostart entry is created for all users. Users are unable to deactivate the autostart function in the AusweisApp. Not specified, no autostart entry is created (false). In that case, users are able to activate the autostart function in the AusweisApp.

TRAYICON Activates the tray icon to keep the AusweisApp active in the background to always be available for an authentication.

AUTOHIDE Concerns the automatic minimization after a successful authentication. Not specified, it is activated (true). Setting AUTOHIDE=false, it is deactivated. Users can adjust this setting to their preferences.

REMINDTOCLOSE Closing the AusweisApp by clicking on the X, the user is notified that only the user interface is closed and that the AusweisApp is still available in the info tray (if the tray icon is enabled) or that the AusweisApp will be shut down and the user needs to restart it to identify towards providers. At this point, it is possible to prevent future notifications. Setting REMINDTOCLOSE=false deactivates this notification from the outset. Not specified, it is activated (true).

[1] https://docs.microsoft.com/en-us/windows/desktop/msi/error-codes
[2] https://docs.microsoft.com/en-us/windows/desktop/msi/standard-installer-command-line-options

ASSISTANT Starting the AusweisApp for the first time, the user interface is displayed and the installation wizard is shown. With each subsequent start, the AusweisApp is started in the background, without the installation wizard being shown. By indicating ASSISTANT=false, the AusweisApp is started in the background without the installation wizard from the outset. Not specified, the installation wizard is activated (true).

TRANSPORTPINREMINDER Prior to the first authentication, the user is asked once whether they have changed their Transport PIN. Setting TRANSPORTPINREMINDER=false deactivates this reminder. Not specified, the reminder is activated (true).

CUSTOMPROXYTYPE Part of a proxy configuration. Valid values are SOCKS5 and HTTP. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYHOST and CUSTOMPROXYPORT). You can disable the proxy after installation with a checkbox in the settings.

CUSTOMPROXYHOST Part of a proxy configuration. Sets the Host of the proxy. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYTYPE and CUSTOMPROXYPORT). You can disable the proxy after installation with a checkbox in the settings.

CUSTOMPROXYPORT Part of a proxy configuration. Sets the port of the proxy. Only values between 1 and 65536 are valid. All proxy parameters have to be set to use the proxy (see CUSTOMPROXYTYPE and CUSTOMPROXYHOST). You can disable the proxy after installation with a checkbox in the settings.

UPDATECHECK On Windows the AusweisApp checks on startup and in a 24 hour interval if a new version is available. If a newer version is available and the tray icon is active then the user is informed via a notification. Additionally a note regarding the new version will be displayed when opening the AusweisApp interface the next time. Setting UPDATECHECK to false or true deactivates or activates the update check respectively. Users are unable to change this setting in the AusweisApp. Not specified, the update check is activated, but users can adjust the settings. The UPDATECHECK parameter affects neither updates of the service provider list nor updates of card reader information.

SHUFFLESCREENKEYBOARD If the on-screen keyboard is activated, the number keys can be arranged at random. By setting SHUFFLESCREENKEYBOARD to false or true, the random arrangement can be deactivated or activated. Users are able to adjust the setting.

SECURESCREENKEYBOARD If the on-screen keyboard is activated, the animation of the number keys can be disabled. By setting SECURESCREENKEYBOARD to false or true, the animation can be activated or deactivated. Users are able to adjust the setting.

ENABLECANALLOWED Enables support for the CAN allowed mode. If the provider got issued a corresponding authorization certificate the ID card can be read by entering the CAN instead of the PIN.

SKIPRIGHTSONCANALLOWED Skips the page with the authorization certificate in the CAN allowed mode and asks directly for the CAN.

LAUNCH Starts the AusweisApp after the installation has finished.

Alternatively, Orca[3] can be used to create an MST file that defines the above arguments. The arguments are available in the "Directory" and "Property" tables. The MST file can be transferred with the following command:

```
msiexec /i AusweisApp-X.YY.Z.msi /quiet TRANSFORMS=file.mst
```

[3] https://docs.microsoft.com/en-us/windows/desktop/Msi/orca-exe

In order to optimize the start of the AusweisApp on systems with no graphics acceleration, the system variable "QT_QUICK_BACKEND" can be set to the value "software". In this case, the AusweisApp does not attempt to use graphics acceleration and starts directly with the alternative software renderer.

1.2 macOS

MacOS does not provide a command line installation. However, some of the above settings can be specified system-wide by a plist file in the /Library/Preferences directory. This plist file must be manually stored by the administrator of the system and will be used by all (future) installations of AusweisApp. All not mentioned settings are not supported on macOS. The name of the file must be "com.governikus.AusweisApp2.plist". The content is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/␣
→PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>trayIcon</key>
  <false/>
  <key>autoCloseWindow</key>
  <false/>
  <key>remindToClose</key>
  <false/>
  <key>showOnboarding</key>
  <false/>
  <key>transportPinReminder</key>
  <false/>
  <key>customProxyType</key>
  <string>HTTP</string>
  <key>customProxyHost</key>
  <string>proxy.example.org</string>
  <key>customProxyPort</key>
  <integer>1337</integer>
  <key>shuffleScreenKeyboard</key>
  <true/>
  <key>visualPrivacy</key>
  <true/>
  <key>enableCanAllowed</key>
  <true/>
  <key>skipRightsOnCanAllowed</key>
  <true/>
</dict>
</plist>
```

The description for each value is applicable for both Windows and macOS, although the naming of the attributes differs, as shown in the following table:

[4] On macOS the AusweisApp is minimized to the menu bar.

Table 1:

macOS	Windows
trayIcon	TRAYICON
autoCloseWindow	AUTOHIDE
remindToClose[4]	REMINDTOCLOSE
showOnboarding	ASSISTANT
transportPinReminder	TRANSPORTPINREMINDER
customProxyType	CUSTOMPROXYTYPE
customProxyPort	CUSTOMPROXYPORT
customProxyHost	CUSTOMPROXYHOST
shuffleScreenKeyboard	SHUFFLESCREENKEYBOARD
visualPrivacy	SECURESCREENKEYBOARD
enableCanAllowed	ENABLECANALLOWED
skipRightsOnCanAllowed	SKIPRIGHTSONCANALLOWED

It might be necessary to force a reload of the data cached by the operating system: `killall -u $USER cfprefsd`

1.3 Operational Environment Requirements

1.3.1 Required authorization for installation and execution

Administrator privileges are required to install the AusweisApp.

The execution of the AusweisApp does not require administrator privileges.

1.3.2 Used network ports

All network ports used by the AusweisApp are listed in Network connections of the AusweisApp (Page 8). Communication model of the AusweisApp (Page 6) shows a schematic representation of the individual connections made by the AusweisApp.

The AusweisApp starts a HTTP-Server on port 24727. The server binds only to the localhost network interface. The availability of the local server is necessary for the online eID function, because providers will redirect the user with a HTTP redirect to the local server to continue the authentication process in the AusweisApp (eID1). The server is also used to offer other local applications to use the AusweisApp via a websocket interface (SDK function, eID-SDK). Therefore local incoming network connections to TCP Port 24727 must be permitted.

If the proxy service is activated, the AusweisApp proxy takes over the server functions of AusweisApp on port 24727. The entities of AusweisApp recognize the proxy and use a free random port in this case to which the proxy forwards the requests.

Broadcast on UDP port 24727 in the local subnet have to be receivable by the AusweisApp to use the "Smartphone as Card Reader" functionality. It may be necessary to deactivate AP isolation on your router.

The installer of the AusweisApp provides an option to register all needed firewall rules in the Windows Firewall. If the rules are not registered, the user will be prompted by the Windows Firewall to allow the outgoing connections once the AusweisApp tries to connect to a server. These prompts are suppressed by registering the firewall rules during installation. No rules have to be added to the Windows Firewall for the local connections eID1 and eID-SDK (when using the standard settings).

In table Firewall rules of the AusweisApp (Page 8) all firewall rules registered by the installer are listed.

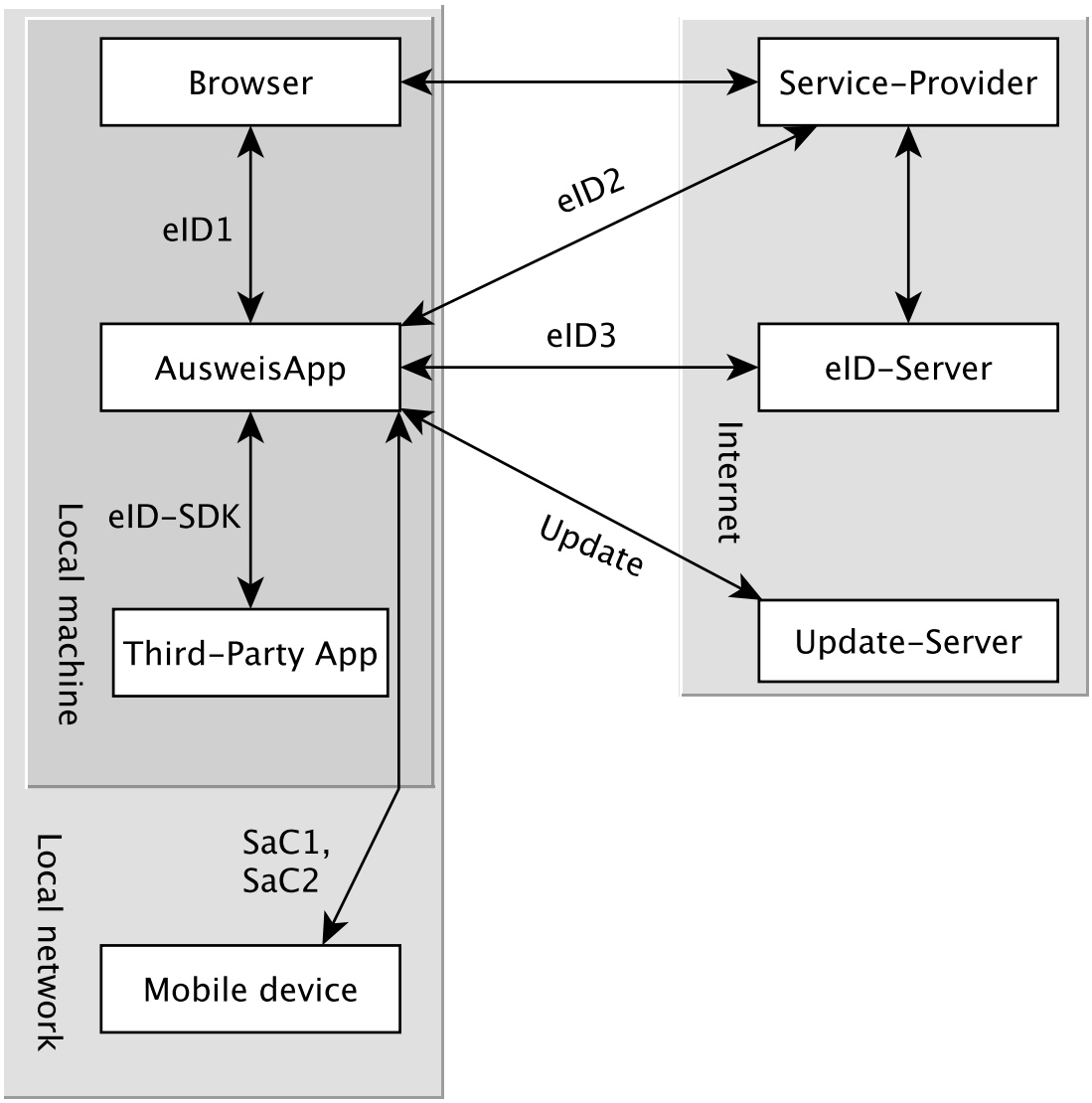

Figure 1: Communication model of the AusweisApp

1.3.3 TLS connections

Transmitted TLS certificates are solely validated via the interlacing with the authorization certificate issued by the german eID PKI. CA certificates in the Windows truststore are thus ignored. It is therefore generally not possible to use the AusweisApp behind a TLS termination proxy.

Table 2: Network connections of the AusweisApp

Reference	Protocol	Port	Direction	Optional	Purpose	Note
eID1	TCP	24727 1[4]	incoming	no	Online eID function, eID activation 2[4]	Only accessible from localhost 2[4]
eID2	TCP	443 3[4]	outgoing	no	Online eID function, connection to the provider, TLS-1-2 channel 2[4]	TLS certificates interlaced with authorization certificate 2[4]
eID3	TCP	443 3[4]	outgoing	no	Online eID function, connection to eID-Server, TLS-2 channel 2[4]	TLS certificates interlaced with authorization certificate 2[4]
eID-SDK	TCP	24727 1[4]	incoming	no	Usage of the SDK functionality	Only accessible from localhost 2[4]
SaC1	UDP	24727 1[4]	incoming	yes	Smartphone as Card Reader, detection 4[4]	Broadcasts
SaC2	TCP		outgoing	yes	Smartphone as Card Reader, usage 4[4]	Connection in local subnet
Update	TCP	443	outgoing	yes	Updates 5[4] of provider and card reader information as well as information on new AusweisApp versions 6[4] .	TLS certificates will be validated against CA certificates included in the AusweisApp. CA certificates provided by the OS are ignored.

Table 3: Firewall rules of the AusweisApp

Name	Protocol	Port	Direction	Connection reference
AusweisApp-Outbound	TCP	$*$	outgoing	eID2, eID3, SaC2, Update
AusweisApp-SaC	UDP	24727	incoming	SaC1

[5] Or a random port when using AusweisApp proxy.
[6] See TR-03124 specification from the BSI
[7] Port 443 is used for the initial contact with the provider or eID server. Due to configuration of the service on the service provider's behalf, any other port might be used by forwarding.
[8] See TR-03112-6 specification from the BSI
[9] All updates are based on the URL https://updates.autentapp.de/
[10] Automatic checks for new AusweisApp versions can be deactivated, see commandline parameter UPDATECHECK.

2 Developer Options

AusweisApp features so-called developer options. They provide advanced settings and facilitate the integration of eID services. The developer options are hidden by default.

2.1 Activating the Developer Options

Developer options are activated by clicking the "Application Version" accessible via "Help" -> "Information" 10 times. After the 10th time, you will receive a notification that the developer options are activated. Once activated, you will find a new category "developer options" in the settings menu. In the mobile versions additional options for "on-site reading" appear.

2.2 Advanced Settings

The developer options offer advanced settings, which are explained below.

2.2.1 Test mode for self-authentication (Test PKI)

In general, the self-authentication is a built-in service of AusweisApp and can only be used with genuine ID cards. However, when in test mode, AusweisApp uses a test service allowing for self-authentication with a test ID card. The test mode (Test PKI) of the self-authentication may be activated or deactivated without using the Developer Options. By clicking on the magnifying glass on the self-authentication start screen ten times, you can enable or disable the test mode.

2.2.2 Internal card Simulator

The internal card simulator allows to run an authentication in the Test PKI without any ID card or card reader. Note that no other card reader can be used in the stationary versions while the simulator is activated.

A single static profile is stored in the current version, which cannot be changed via the graphical user interface. Only the SDK allows to change the profile's data using the SET_CARD command. Further information can be found at the documentation of AusweisApp SDK (see Software Development Kit (SDK) (Page 10)).

2.2.3 Developer mode (stationary only)

When the developer mode is activated, some safety measures during an authentication process are ignored. Ignoring the safety measures with test services usually employed in test scenarios, yields a successful authentication. Each safety breach will be highlighted as an internal notification in AusweisApp or the operating system respectively.

The following safety tests are disabled in the developer mode:

- The used TLS keys and ephemeral TLS keys have the necessary minimum length.
- The URL of the TLS certificate description of the eID server and the TcToken URL must fulfill the same-origin policy.
- The used TLS certificates must be entwined with the authorization certificate.
- The RefreshAddress URL and possible redirect URLs must conform to the HTTPS scheme.

Please note: Developer mode can only be used for test services, usage with genuine provider certificates is not possible.

2.2.4 Show notifications inside the app

With this option, notifications are displayed within the app instead of as system notifications. When enabled, a bell icon appears in the top right corner of the application window. Clicking the bell icon shows past notifications with timestamps. This option is automatically enabled when the developer mode is active, as security violations are listed there.

2.2.5 Support CAN Allowed mode for on-site reading (mobile only)

Enables support for the CAN allowed mode. If the provider got issued a corresponding authorization certificate the ID card can be read by entering the CAN instead of the PIN.

2.2.6 Skip rights page

Skips the page with the authorization certificate in the CAN allowed mode and asks directly for the CAN.

3 Software Development Kit (SDK)

3.1 Possible Uses

The software development kit (SDK) of AusweisApp enables you to integrate the eID function directly into your own application or app. This enables users to authenticate themselves without media discontinuity.

The SDK offers the advantage of being able to carry out an online authentication in your own brand design - without users having to leave the familiar environment.

The AusweisApp SDK also enables the integration of on-site reading. In this case, the CAN is transmitted instead of the PIN to enable data transmission. You find the CAN on the front of the ID card and you need it to enable the readout process.

3.2 Integration Options

With the fully integrated version of the SDK, AusweisApp is integrated into your own application as an AAR package or Swift package. The advantage: AusweisApp is delivered directly with the application so that users don't have to install AusweisApp separately on their smartphone.

With the partially integrated version of the SDK, AusweisApp is called in the background. Where applicable, however, the app can be delivered with the installer regardless of partial integration.

Table 4:

	partially integrated	fully integrated
Windows / macOS	Yes	No
Android	No	Yes
iOS	No	Yes

3.3 Developer documentation

You can find a detailed developer documentation of the SDK with a list of possible failure codes at https://www.ausweisapp.bund.de/sdk/.

3.4 SDK Wrapper

You can use the SDK Wrapper of the AusweisApp to simplify the integration of the SDK into your app. The SDK Wrapper offers Swift and Kotlin bindings for iOS and Android.