

# Tungsten SignDoc Standard Installation Guide

Version: 3.4.0

Date: 2024-11-15

**TUNGSTEN**  
**AUTOMATION**

© 2014–2024 Tungsten Automation. All rights reserved.

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

# Table of Contents

|  |           |
|--|-----------|
| <b>Preface</b> .....   | <b>5</b>  |
| Product documentation.....                                   | 5         |
| Offline documentation.....                                   | 6         |
| Training.....  | 7         |
| Getting help with Tungsten Automation products.....          | 7         |
| <b>Chapter 1: Introduction</b> .....                         | <b>9</b>  |
| General overview.....  | 9         |
| Non-Goals of this guide.....                                 | 9         |
| Reverse proxy / Load balancing / SSL.....                    | 9         |
| Contents of SignDoc Standard.....                            | 10        |
| Server configuration.....                                    | 11        |
| <b>Chapter 2: Base installation</b> .....                    | <b>14</b> |
| Installation as a Windows service.....                       | 14        |
| Quick start.....   | 14        |
| Production setup.....  | 15        |
| Configure the External Host URL.....                         | 16        |
| Configure the database connection.....                       | 16        |
| Configure network settings.....                              | 17        |
| Configure SMTP server connection.....                        | 18        |
| Tungsten TotalAgility integration.....                       | 18        |
| Configure LDAP integration.....                              | 19        |
| Windows Authentication for database connection.....          | 21        |
| Monitoring.....  | 21        |
| Cookies.....   | 22        |
| Uninstall SignDoc Standard.....                              | 23        |
| Create accounts and users.....                               | 23        |
| <b>Chapter 3: Advanced installation</b> .....                | <b>24</b> |
| Hardening a SignDoc installation.....                        | 24        |
| Content Security Policy (CSP).....                           | 25        |
| <b>Chapter 4: Upgrade SignDoc Standard</b> .....             | <b>27</b> |
| Upgrade from SignDoc Standard 3.3.1 or earlier versions..... | 27        |
| Upgrade from SignDoc Standard 2.2.1.....                     | 28        |
| Upgrade troubleshooting.....                                 | 29        |
| <b>Chapter 5: Database migration</b> .....                   | <b>30</b> |
| Database migration with Flyway.....                          | 30        |

|  |           |
|--|-----------|
| Load balancing considerations.....         | 30        |
| <b>Chapter 6: Troubleshooting.....</b>     | <b>32</b> |
| General.....                               | 32        |
| Sanity Check.....                          | 32        |
| SignDoc Standard does not start.....       | 32        |
| SignDoc Standard does not send emails..... | 33        |
| <b>Appendix A: Glossary.....</b>           | <b>34</b> |

# Preface

This guide contains information that administrators need to install or upgrade Tungsten SignDoc Standard.

For information on supported operating systems and other system requirements, refer to the [Tungsten SignDoc Technical Specifications](#) document.

This document is updated regularly, and we recommend that you review it carefully to ensure success with your Tungsten SignDoc product.

## Product documentation

The full documentation set for SignDoc Standard is available at the following location:

<https://docshield.tungstenautomation.com/Portal/Products/SD/3.4.0-rq5j77n6cd/SD.htm>

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

### Release notes

- *Tungsten SignDoc Release Notes*

### Technical specifications

- *Tungsten SignDoc Technical Specifications*

### Guides

- *Tungsten SignDoc Standard Administrator's Guide*
- *Tungsten SignDoc Standard Developer's Guide*

### Help

- *Tungsten SignDoc Standard Help*
- *Tungsten SignDoc Standard Administration Center Help*
- *Tungsten SignDoc Assistant Help*
- *Tungsten SignDoc Help - Signing Documents*
- *Tungsten SignDoc Device Connector Help*
- *Tungsten SignAlyze Help*

### Software development kit

- *Tungsten SignDoc Browser Capture Help*
- *Tungsten SignDoc SDK API Documentation (C)*
- *Tungsten SignDoc SDK API Documentation (C++)*
- *Tungsten SignDoc SDK API Documentation (.NET with exceptions)*


- *Tungsten SignDoc SDK API Documentation (.NET without exceptions)*
- *Tungsten SignDoc SDK API Documentation (Java)*

## Offline documentation

Customers who require offline documentation can download the English documentation package `TungstenSignDocDocumentation_3.4.0_EN.zip` from the [Tungsten Automation Fulfillment](#) site. The .zip file includes the `help` directory where you can find the SignDoc help files and the `print` directory with the SignDoc guides.

### Tungsten SignDoc Standard help

The following steps describe how to make the English offline help accessible in SignDoc Standard (Administration Center, Manage Client, and Signing Client) by copying the help to the internal web server (Tomcat) for the installation.

 Before proceeding, you must install SignDoc Standard in the directory `<INSTALLDIR>` and set the `<SERVICE_EXTERNAL_HOST_URL>` as described in this guide.

1. From the Tungsten Automation Fulfillment site, download `TungstenSignDocDocumentation_3.4.0_EN.zip`.
2. Extract the contents of the .zip file to any directory `<EXTRACTDIR>`.
3. Copy `<EXTRACTDIR>/help` to the directory `<INSTALLDIR>/service/webapp`.
4. Start SignDoc Standard and configure the help links in the System settings of the Administration Center.
  - a. Manage Client  
Open the subcategory Client > Manage and edit "Manage Client online help URL" by entering the URL `<SERVICE_EXTERNAL_HOST_URL>/help/Standard/index.html`.
  - b. Signing Client  
Open subcategory Client > Signing and edit "Signing Client online help URL" by entering the URL `<SERVICE_EXTERNAL_HOST_URL>/help/StandardSigningDocuments/index.html`.
  - c. Administration Center  
Open subcategory Client > Administration and edit "Administration Center online help URL" by entering the URL `<SERVICE_EXTERNAL_HOST_URL>/help/StandardAdministrationCenter/index.html`.
5. Test the configured links by clicking the Help link in the header of Administration Center, Manage Client and Signing Client. Each help system should display in a new browser tab.

### Tungsten SignDoc Standard guides

From the directory `<EXTRACTDIR>/print`, you can access the following guides:

- *Tungsten SignDoc Standard Administrator's Guide*  
`TungstenSignDocStandardAdministratorsGuide_EN.pdf`
- *Tungsten SignDoc Standard Developer's Guide*  
`TungstenSignDocStandardDevelopersGuide_EN.pdf`

- *Tungsten SignDoc Standard Installation Guide*  
TungstenSignDocStandardInstallationGuide\_EN.pdf

### SignDoc Software Developer Kit documentation

According to the functionality and the programming language, the offline documentation .zip file contains documentation for the SignDoc Software Developer Kit.

To open and use the SignDoc Software Developer Kit documentation, follow these steps:

1. From the Tungsten Automation Fulfillment site, download  
TungstenSignDocDocumentation\_3.4.0\_EN.zip.
2. Extract the contents of the .zip file to any directory <EXTRACTDIR>.
3. Navigate to <EXTRACTDIR>/help to access the SignDoc Software Developer Kit documentation.


## Training

Tungsten Automation offers both on-demand and instructor-led training to help you make the most of your product. To learn more about training courses and schedules, visit the [Tungsten Automation Learning Cloud](#).

## Getting help with Tungsten Automation products

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

To access the Tungsten Automation Knowledge Portal, go to <https://knowledge.tungstenautomation.com/>.

 The Tungsten Automation Knowledge Portal is optimized for use with Google Chrome, Mozilla Firefox, or Microsoft Edge.

The Tungsten Automation Knowledge Portal 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.  
To locate articles, go to the Knowledge Portal home page and select the applicable Solution Family for your product, or click the View All Products button.

From the Knowledge Portal home page, you can:

- Access the Tungsten Automation Community (for all customers).  
On the Resources menu, click the **Community** link.
- Access the Tungsten Automation Customer Portal (for eligible customers).

Go to the [Support Portal Information](#) page and click **Log in to the Customer Portal**.

- Access the Tungsten Automation Partner Portal (for eligible partners).

Go to the [Support Portal Information](#) page and click **Log in to the Partner Portal**.

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

Go to the [Support Details](#) page and select the appropriate article.



## Chapter 1

# Introduction

This guide includes instructions for installing and upgrading Tungsten SignDoc Standard, along with information on authentication and database installation, configuration, and migration.

## General overview

### Operating systems

SignDoc Standard can be installed on Windows and Linux operating systems with a 64-bit architecture.

### SignDoc Standard application

Since version 3.4.0 SignDoc Standard is a standalone application that uses an embedded Apache Tomcat as its application server. Apache Tomcat is widely used within the industry. The SignDoc Standard application can be deployed using container technologies.

### REST interface

SignDoc Standard can interact with the system via the REST API, which supports almost all aspects of the application. With a single REST request, creating and scheduling signing packages is possible. For details, refer to the [Tungsten SignDoc product documentation](#) page.

## Non-Goals of this guide

This *SignDoc Standard Installation Guide* does not include descriptions of the following topics:

- Cover the detailed setup of a reverse proxy or load balancing
- Consider or discuss a production-ready SSL configuration, which usually depends on local IT regulations. We recommend offloading SSL using a reverse proxy or load balancer.

## Reverse proxy / Load balancing / SSL

Because SignDoc Standard uses a stateless REST Interface, it can be run behind a reverse proxy or load balancer. Even the simplest load balancing strategies such as round-robin can be used.

The reverse proxy or load balancer can be used to offload SSL and distribute requests to multiple and identical configured SignDoc Standard instances. We generally recommend using an TLS enabled reverse proxy or load balancer to provide a single entry point for the application.

If no reverse proxy is planned to be used, a sample SSL configuration for the SignDoc Standard Service can be found in `SignDocStandard.yml`.


## Contents of SignDoc Standard

When you unzip the SignDoc zip, you will find this directory and file structure:

### Directories

| Directory name            | Description  |
|---------------------------|--|
| <code>jre</code>          | The java runtime   |
| <code>lib</code>          | Shared libraries required at runtime   |
| <code>modules</code>      | SignDoc extension modules  |
| <code>mssql</code>        | Sample scripts to initialize an MS-SQL database  |
| <code>signdoc_home</code> | Contains resources such as fonts files, interfaces documentation and the server plugin installation directory. |

### Files

| File name                            | Description  |
|--------------------------------------|--|
| <code>DockerReadme_linux.md</code>   | Readme file for using Docker Linux containers  |
| <code>DockerReadme_windows.md</code> | Readme file for using Docker Windows containers  |
| <code>Dockerfile_linux</code>        | Sample Docker Linux base image   |
| <code>Dockerfile_linux_mssql</code>  | Sample Docker Linux MS-SQL image   |
| <code>Dockerfile_windows</code>      | Sample Docker Windows base image   |
| <code>SignDocStandard.config</code>  | Configuration for the Windows Service wrapper. Do not change this file.  |
| <code>SignDocStandard.exe</code>     | The SignDoc Standard Service Wrapper   |
| <code>SignDocStandard.yml</code>     | Main configuration file for the SignDoc Standard Service. The contents of this file can be changed and adjusted.   |
| <code>application.properties</code>  | Additional configuration file that can be used to set a base configuration. Configuration in <code>SignDocStandard.yml</code> will overwrite settings done in <code>application.properties</code> .  |
| <code>logback-spring.xml</code>      | Default Logback logging configuration that writes log lines to the console (stdout) and in a file.<br><br><div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;">  This file will be created after first startup. </div> |

| File name          | Description   |
|--------------------|---|
| service_remove.cmd | Windows batch file that stops and removes the SignDoc Standard service. Run as administrator to minimize the Windows Process Elevation Dialogs. |
| service_up.cmd     | Windows batch file that (re)starts the SignDoc Standard service. Run as administrator to minimize the Windows Process Elevation Dialogs.        |
| signdoc-boot.jar   | The SignDoc Standard application server   |
| version.txt        | Version file  |

## Server configuration

For the most part, SignDoc is configured at run-time via the Administration Center or in the Account Configuration. However, a few settings must be applied when starting the server. See "Configure the database connection" and "Configure the external Host URL" in the [Production setup](#) section. Generally, settings can be applied with these approaches in order of increasing precedence:

Options for applying server configuration:

- `application.properties`
- Environment variables (Windows: can be set in `SignDocStandard.yml`)
- Java system properties (Windows: can be set in `SignDocStandard.yml`)
- Application arguments (Windows: can be set in `SignDocStandard.yml`)

We recommend using environmental variables, as they are flexible and do not require sensitive credentials to be stored in files.

### **application.properties**

Configuration settings (properties) are added line by line and usually use a dot-Syntax. See the `application.properties` file for examples.

Example:

```
# A local MSSQL Database connection
#spring.datasource.url=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;
#spring.datasource.username=signdoc
#spring.datasource.password=2beChanged!
```

### **Environment variables**

Environment variables are flexible and especially useful when running SignDoc as a containerized application. The configurations are the same as in `application.properties`, but are written in all uppercase and use the underscore character instead of the dot.

Example (as used in `SignDocStandard.yml`):

```
env:
- name: SPRING_DATASOURCE_URL
  value: 'jdbc:sqlserver://localhost:1433;database=signdoc_34;encrypt=false;'
- name: SPRING_DATASOURCE_USERNAME
  value: 'signdoc'
```

```
- name: SPRING_DATASOURCE_PASSWORD
  value: '2beChanged!'
```

Example (in Dockerfile syntax):

```
# A local MSSQL Database connection
env SPRING_DATASOURCE_URL=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;
env SPRING_DATASOURCE_USERNAME=signdoc
env SPRING_DATASOURCE_PASSWORD=2beChanged!
```

Example (as Docker run parameters):

```
docker run --rm \
  -e 'SPRING_DATASOURCE_URL=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;' \
  -e 'SPRING_DATASOURCE_USERNAME=signdoc' \
  -e 'SPRING_DATASOURCE_PASSWORD=2beChanged!' \
  ubuntu bash -c export
```

Example (in bash syntax):

```
# A local MSSQL Database connection
export SPRING_DATASOURCE_URL=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;
export SPRING_DATASOURCE_USERNAME=signdoc
export SPRING_DATASOURCE_PASSWORD='2beChanged!'
```

## Java system properties

Alternatively, a configuration property can be set as a Java system property.

Syntax: Use the same property names as in `application.properties` and precede them with a minus-D (-D)

Example (as used in `SignDocStandard.yml`):

```
arguments: >
  -Dfile.encoding=UTF-8
  -Djava.library.path="%BASE%\lib"
  -Dspring.datasource.url=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;
  -Dspring.datasource.username=signdoc
  -Dspring.datasource.password=2beChanged!
  -jar
  signdoc-boot.jar
```

## Application arguments

Finally, it is possible to set configuration using application arguments. This method has the highest precedence and overwrites the value of the same property set with one of the above methods.

Syntax: Use the same property names as in `application.properties` and precede them with a double-minus (--)

Example (as used in `SignDocStandard.yml`):

```
arguments: >
  -Dfile.encoding=UTF-8
  -Djava.library.path="%BASE%\lib"
  -jar
```

```
signdoc-boot.jar
--spring.datasource.url=jdbc:sqlserver://
localhost:1433;database=signdoc_34;encrypt=false;
--spring.datasource.username=signdoc
--spring.datasource.password=2beChanged!
```

## Chapter 2

# Base installation

This guide explains how to install Tungsten SignDoc Standard on a Windows 64-bit operating system. Sample screens in this guide may vary from your installation, depending on your version of Windows. For an installation on Linux systems, usage of Docker is recommended.

The installed system always contains these components:

- Application Server (SignDoc Standard)
- Database (Microsoft SQL Server or H2 for testing/demo)

SignDoc Standard is run as an Application Server that uses an embedded Apache Tomcat instance. The SignDoc Standard application server and Microsoft SQL Server are usually installed on different computers for various reasons, but they can technically also be installed on the same computer.

## Installation as a Windows service

This section provides information on installing SignDoc Standard as a Windows service.

### Definitions

- <INSTALLDIR> is the directory of the unpacked `signdoc_standard*-tomcat.zip` file.

### General prerequisites

Before starting the installation, you must verify that your system meets the requirements in the [Tungsten SignDoc Technical Specifications](#) document.

If you plan to upgrade an existing SignDoc Standard installation, see [Upgrade SignDoc Standard](#).


**i** We recommend installing SignDoc Standard behind a reverse proxy. If the reverse proxy is also used for load balance requests, they can be handled as stateless, such as round robin.

## Quick start

Getting a simple local accessible SignDoc Standard installation running can be achieved in less than 5 minutes. The local installation can serve as a base for a production-ready setup.

### Quick start goals


- Install SignDoc Standard as a Windows service.
- Database: preconfigured for a local file-based H2 database.

 The data stored in the local file-based H2 database cannot be migrated to a production database based on Microsoft SQL Server.

- SMTP configuration: preconfigured for localhost with port 1025 (no authentication or encryption).
  - Works with [MailHog](#) (email testing tool).
  - MailHog can be stopped and deleted at any time. After a working SMTP server is configured, MailHog is not needed.

### Quick start prerequisites

- 8 GB RAM
- Run MailHog after you download it from the GitHub website.  
If this is not possible or not preferred, an SMTP server must be configured first. See [Configure SMTP server connection](#).
- Access MailHog: `http://localhost:8025` (if applicable).
- Enable startup email feature. See "Startup email" in the [Production setup](#) section.

 MailHog is only needed for the quick start scenario. When using startup email with a valid SMTP server, make sure to use a real email address.

### Quick start procedure

1. Double-check that the general prerequisites are fulfilled.
2. Extract `signdoc-standard*-tomcat.zip` in a new directory `<INSTALLDIR>`.  
Example:  
`C:\Program Files\signdoc-standard-3.4.0`
3. Double-click `<INSTALLDIR>\service_up.cmd`. You will have to approve Windows 4 elevation dialogs.
4. Wait approximately 1 minute on first start.
5. A SignDoc Standard startup email should be sent to the specified startup email recipient while starting up. Ensure that the `cirrus.startup.email` property is not commented in `application.properties`.
6. Open SignDoc Standard: `http://localhost:6611`.
7. Log in using the first log-in credentials (refer to [First login](#) in the *Tungsten SignDoc Standard Administration Center Help*).

## Production setup

The following sections describe basic tasks for a production setup.

### Goals for production

- [Configure the External Host URL \(required\)](#)
- [Configure database connection \(required\)](#)

- [Configure network settings \(optional\)](#)
- [Configure SMTP server connection \(optional\)](#)
- [Tungsten TotalAgility integration \(optional\)](#)
- [Configure LDAP integration \(optional\)](#)
- [Monitoring \(optional\)](#)

### Prerequisites for production setup

- Application server
  - Minimum 16 GB RAM.
  - Minimum 5 GB free disk space
- Database
 

Installed Microsoft SQL Server with a database for SignDoc Standard and a database user with database owner (dbo) credentials for this database. See [Prepare Microsoft SQL database](#).
- SignDoc Standard
 

Install SignDoc Standard as described in [Quick start](#).


### Procedure for production setup

The following topics do not depend on each other and can be executed independently. What is common for all settings: The settings must be applied by executing `<INSTALLDIR>\service_up.cmd`.

## Configure the External Host URL

This is a required setting.

The application must be aware of the externally accessible Host URL. This setting is especially important when SignDoc is run behind a reverse proxy server.

 This URL must not include the application context (i.e., /cirrus).

Example:

```
service.external.host.url=http://myserver:6611
```

As environment variable

```
SERVICE_EXTERNAL_HOST_URL=http://myserver:6611
```

## Configure the database connection

This is a required setting.

The database connection is configured using a JDBC connection string.

An H2 database connection can be used for testing or demo purposes, but only MS-SQL is supported for production usage.

Example 1: In-memory H2 with persistent file-storage



```
spring.datasource.username=signdoc
spring.datasource.password=2beChanged!
spring.datasource.url=jdbc:h2:signdoc_database;DB_CLOSE_DELAY=-1;INIT=SET COLLATION
ENGLISH STRENGTH SECONDARY
```

#### As environment variables

```
SPRING_DATASOURCE_USERNAME=signdoc
SPRING_DATASOURCE_PASSWORD=2beChanged!
SPRING_DATASOURCE_URL="jdbc:h2:signdoc_database;DB_CLOSE_DELAY=-1;INIT=SET COLLATION
ENGLISH STRENGTH SECONDARY"
```

#### Example 2: MS-SQL connection

```
spring.datasource.username=signdoc
spring.datasource.password=2beChanged!
spring.datasource.url=jdbc:sqlserver://localhost:1433;database=signdoc;encrypt=false;
```

#### As environment variables

```
SPRING_DATASOURCE_USERNAME=signdoc
SPRING_DATASOURCE_PASSWORD=2beChanged!
SPRING_DATASOURCE_URL=jdbc:sqlserver://localhost:1433;database=signdoc;encrypt=false;
```

## Configure network settings

This is an optional setting.

The network settings can be configured to bind the application to a specific network interface or IP address

Define HTTP port

Example:

```
server.port=6611
```

As environment variable

```
SERVER_PORT=6611
```

SSL configuration

Example:

```
server.ssl.enabled=true
server.ssl.key-store=./keystore.p12
server.ssl.key-store-password=2beChanged!
server.ssl.key-store-type=PKCS12
server.ssl.key-alias=1
```

As environment variables

```
SERVER_SSL_ENABLED=true
SERVER_SSL_KEY_STORE=./keystore.p12
SERVER_SSL_KEY_STORE_PASSWORD=2beChanged!
SERVER_SSL_KEY_STORE_TYPE=PKCS12
SERVER_SSL_KEY_ALIAS=1
```

## Configure SMTP server connection

This is an optional setting.

**i** The default SMTP server can also be configured in the Administration Center, so this chapter is optional. See also the SMTP section of the Administration Center for possible properties.

### Example SMTP configuration

```
mail.smtp.host=email-smtp.us-east-1.amazonaws.com
mail.smtp.port=587
mail.smtp.user=<Access key ID>
mail.smtp.from=dont_reply@mydomain.com
mail.smtp.password=<Secret access key>
mail.smtp.starttls.enable=true
mail.smtp.starttls.required=true
mail.smtp.ssl.checkserveridentity=false
```

### As environment variables

```
MAIL_SMTP_HOST=email-smtp.us-east-1.amazonaws.com
MAIL_SMTP_PORT=587
MAIL_SMTP_USER=<Access key ID>
MAIL_SMTP_FROM=dont_reply@mydomain.com
MAIL_SMTP_PASSWORD=<Secret access key>
MAIL_SMTP_STARTTLS_ENABLE=true
MAIL_SMTP_STARTTLS_REQUIRED=true
MAIL_SMTP_SSL_CHECKSERVERIDENTIRY=false
```

### Startup email

It can be useful to send a startup email to a predefined address whenever the SignDoc Standard server is starting. SignDoc Standard can be configured for this purpose. The startup email contains information about configuration, the environment, and start parameters for the SignDoc Standard application. To enable the startup email, configure the `cirrus.startup.email` setting with one of the options listed in [Server configuration](#).

Example:

```
cirrus.startup.email=signdoc-startup-email@example.com
```

As environment variable

```
CIRRUS_STARTUP_EMAIL=signdoc-startup-email@example.com
```

## Tungsten TotalAgility integration

This is an optional setting.

The Tungsten TotalAgility connection is individually defined per account/tenant in the SignDoc Standard Manage Client or SignDoc Standard Administration Center.

For details, refer to the "Plug-ins" topic in the *Tungsten SignDoc Standard Administration Center Help* and "KTA state change plug-in" section in the *Tungsten SignDoc Standard Administrator's Guide*

## Configure LDAP integration

This is an optional setting.

LDAP (or Active Directory) can be used to authenticate SignDoc Standard users in the Manage Client. The LDAP support in SignDoc Standard is not usable and not supported in a multi-tenant environment. SignDoc Standard maps LDAP user entries to SignDoc Standard users by the unique email address. A SignDoc Standard multi-tenant installation requires email addresses only to be unique within a single account.

The user's ID is defined by the setting `ldap.user.mail.attr`. It must represent the email address of the user. This is not a standard LDAP attribute and may have to be added by a system administrator.

### Activating LDAP

LDAP support is activated by setting the property `authentication.provider` to the value `LDAP,CIRRUS`. Activate LDAP only after you have created your first account in SignDoc.

Example:

```
authentication.provider=LDAP,CIRRUS
```

As environment variable

```
AUTHENTICATION_PROVIDER=LDAP,CIRRUS
```

### Auto creating a user

If a user logs in and a user with the mail address received from LDAP does not exist, a user is created automatically in SignDoc Standard. SignDoc Standard maps LDAP attributes to SignDoc Standard user attributes. Each name of an LDAP attribute has a default value but can be customized by a SignDoc Standard property:

| SignDoc Standard property / Environment variable       | Default value | Mapped to this SignDoc Standard user attribute | Constraints  |
|--|---------------|--|--|
| <code>ldap.user.name.attr / LDAP_USER_NAME_ATTR</code> | cn            | user name                                      |  |
| <code>ldap.user.mail.attr / LDAP_USER_MAIL_ATTR</code> | mail          | OID  | (mandatory setting)<br>Must not already exist as a SignDoc Standard user. The email address must match this regular expression:<br><code>^[A-z0-9\.\_%+\-]+@[A-z0-9\.\-]+</code> |

| SignDoc Standard property / Environment variable | Default value | Mapped to this SignDoc Standard user attribute | Constraints   |
|--|---------------|--|---|
| ldap.user.uid.attr / LDAP_USER_UID_ATTR          | uid           | user name                                      | (optional setting) Must not already exist and match the SignDoc Standard validation rule for OID (regex <code>^[a-zA-Z0-9\\._\\-]+\$</code> ) |

If one of the above constraints are violated, SignDoc Standard will report an error and LDAP integration will not work reliably.

### LDAP configuration

| SignDoc Standard property / Environment variable  | Description   | Example   |
|---|---|---|
| authentication.provider / AUTHENTICATION_PROVIDER | Activates LDAP support. Must be LDAP,CIRRUS   |   |
| ldap.url / LDAP_URL                               | The URL to connect to an LDAP server. It's recommended using the ldaps protocol because passwords are sent in plain text over the network. In this situation, a suitable certificate must be installed. | ldap://ad.kofax.com:389/dc=kofax,dc=de                          |
| ldap.manager.dn / LDAP_MANAGER_DN                 | The manager DN. If your LDAP implementation does not allow anonymous access, a suitable user and password must be defined here  | uid=admin,ou=system   |
| ldap.manager.password / LDAP_MANAGER_PASSWORD     | The manager password  | ldap.manager.password=secret                                    |
| ldap.userdn.patterns / LDAP_USERDN_PATTERNS       | A list of distinguished names (DN) separated by a colon.  | uid={0},ou=Users The key {0} is substituted with the login name |
| ldap.user.search.base / LDAP_USER_SEARCH_BASE     | The base DN for starting a search   | dc=kofax,dc=de  |
| ldap.user.search.filter / LDAP_USER_SEARCH_FILTER | A filter for the search (see RFC 2254)  | (cn=Babs Jensen)  |
| ldap.user.name.attr / LDAP_USER_NAME_ATTR         | The LDAP attribute that maps to a SignDoc Standard user name. Default: cn   | ldap.user.name.attr=cn  |
| ldap.user.mail.attr / LDAP_USER_MAIL_ATTR         | The LDAP attribute which maps to a SignDoc Standard user email. Default:  | ldap.user.mail.attr=mail  |

**i** Additional "Brute Force Authentication Prevention" is not implemented if LDAP Authentication is configured.

## Windows Authentication for database connection

SignDoc supports Windows Authentication for the SQL Server connection.

SignDoc uses the jdbc SQL Driver from Microsoft to connect to the SQL Server.

### Prerequisites for implicit Windows Authentication

- Download the Microsoft JDBC Driver 12.6 for SQL Server from [Microsoft](#) and copy the `sqljdbc_auth.dll` into the `lib` directory of the SignDoc installation.
- The SignDoc service must be started under a Windows account, which can be used for this Windows Authentication. This can be configured in the `SignDocStandard.yml` file.

Please check [Building the connection URL](#) for more details.

For Windows Authentication you need to add `integratedSecurity=true` to the query string of `spring.datasource.url`.

Example:

```
jdbc.url=jdbc:sqlserver://  
SERVER:1433;databaseName=signdoc;encrypt=false;integratedSecurity=true
```

This JDBC URL will use Windows Authentication for the SQL Server connection.

## Monitoring

SignDoc Standard can be monitored using the following methods:

- Simple "is alive" endpoint: `/pong`

Example:

```
http://localhost:6611/cirrus/pong
```

- JMX monitoring:

SignDoc Standard exposes the SignDoc JMX bean next to the standard JVM and Spring JMX beans for monitoring. The JMX beans can be accessed using JMX clients like JConsole or VisualVM.

The SignDoc JMX MBean provides global status information about the SignDoc Standard service:

- `Accounts`: All account IDs
- `ActiveSigningSessions`: Number of active signing sessions
- `SysAdmins`: All system administrator usernames
- `TotalAccounts`: Number of all accounts of all accounts
- `TotalDocuments`: Number of all documents of all accounts
- `TotalSigningPackages`: Number of all signing packages of all accounts
- `TotalUsers`: Number of all users of all accounts

## Cookies

SignDoc needs to make use of HTTP Cookies for certain use cases like for example SSO.

Please make sure that the infrastructure SignDoc is installed does not delete cookies set by the SignDoc Server application.

### Test Environment / Infrastructure

To test if HTTP Cookies work as required, open the URL

```
http://<server>/cirrus/test/cookie
```

Example:

```
http://localhost:6611/cirrus/test/cookie
```

This endpoint will set a cookie in the response and do a 302 redirect to:

```
/cirrus/test/cookie/result
```

The cookie is created using these parameters:

- Name: COOKIE-TEST-<epoch\_seconds>
- Value: epoch\_seconds as readable ISO formatted string
- Max-Age: 60 seconds

If the test is successful, the result page will respond with:

Cookie Test => Passed!  <some more cookie details>

If the test failed, the result page will respond with:

Cookie Test => Failed!  <some more cookie details, usually "n/a=n/a">

Note that the cookie expires after 60 seconds. Reloading the result page after this timeframe will yield to a failed result. In this case, simply restart the test with:

```
/cirrus/test/cookie
```

### Logfile

The test results will be printed using INFO level.

### Disabling the endpoints

To disable the endpoints, set this ENV variable:

```
CIRRUS_TEST_ENDPOINTS_ENABLE=false
```

Alternatively, put this setting in `application.properties`:

```
cirrus.test.endpoints.enable=false
```

Restart the SignDoc Service after applying the changes.

## Uninstall SignDoc Standard

To uninstall SignDoc Standard

1. Double-click `<INSTALLDIR>\service_remove.cmd`. Follow the prompts and wait until the Windows service "SignDoc Standard" is stopped and deregistered.
2. Delete the installation directory `<INSTALLDIR>`.

## Create accounts and users

Open the Administration Center to create accounts and users:

```
http[s]://<server>/cirrus/admin-center
```

In this guide this is:

```
http://<SERVICE_EXTERNAL_HOST_URL><SERVICE_HTTP_PORT>/cirrus/admin-center
```

See [Production setup](#).

Example:

When `<SERVICE_EXTERNAL_HOST_URL>` is `localhost` and `SERVICE_HTTP_PORT` is `6611` the URL is:

```
http://localhost:6611/cirrus/admin-center
```

To create accounts and users, refer to the *Tungsten SignDoc Standard Administration Center Help*.

## Chapter 3

# Advanced installation

After you have set up SignDoc Standard as described in the base installation, you can continue with the procedures in this chapter, if necessary.

## Hardening a SignDoc installation

Hardening a SignDoc setup means to apply best practices and security measures to a SignDoc installation for production usage.

### Reverse proxy

We recommend running SignDoc behind a reverse proxy, since this provides an additional abstraction layer between the application server and the users. The reverse proxy:

- Can act as a TLS/SSL endpoint for the system, which simplifies deployment and maintenance.
- Is usually capable of load-balancing requests to multiple SignDoc installations, which improves the high-availability of an installation.
- Can be configured to set specific HTTP header attributes to minimize commonly known attack vectors such as XSS.

### HTTP headers

Additional HTTP response headers can be set or added using the configuration options `security.http.response.headers.set` or `security.http.response.headers.add` of the Administration Center.

Make sure to set these HTTP headers concerning security:

```
X-Frame-Options "SAMEORIGIN";  
X-Content-Type-Options "nosniff";  
X-XSS-Protection "1; mode=block";
```

### Turn off server tokens

Application servers and reverse proxies often announce their identity and version via server tokens in the HTTP response. This information is superfluous and should be avoided.

### Block excessively large uploads

The reverse proxy should reject upload requests that are too big. But it must also allow uploads that are justified. Some reverse proxies have a very small upload limit that must be increased. The limit must be at least 33% (recommended 50%) larger than the size of a single document to be uploaded.



Example: If the largest acceptable document size is 60 MB, the upload limit (maximum body size) 90 MB (+50%) would be a safe limit.

## TLS/SSL

When the reverse proxy acts as a TLS endpoint, it must reject unsafe or outdated SLL protocols or cipher versions. Also, the HTTP protocol should be disabled.

## Block access to URIs

Some resources are not needed for a typical production environment and can be safely blocked:

- /cirrus/swagger
- /cirrus/api-docs
- /cirrus/static/swagger-ui

## Example configurations

The following files show an example configuration for a Nginx web server.

File: my\_proxy.conf

```
server_tokens off;
client_max_body_size 100m;
```

File: default

```
add_header X-Frame-Options "SAMEORIGIN";
add_header X-Content-Type-Options "nosniff";
add_header X-XSS-Protection "1; mode=block";

location cirrus/swagger {
    return 404;
}
location cirrus/api-docs {
    return 404;
}
location cirrus/static/swagger-ui {
    return 404;
}
location /cirrus/swagger {
    return 404;
}
location /cirrus/api-docs {
    return 404;
}
location /cirrus/static/swagger-ui {
    return 404;
}
```

## Content Security Policy (CSP)

SignDoc automatically sets a default Content-Security-Policy HTTP header depending on the user interface being active in the browser. The default Content-Security-Policy header values can be found in the Administration Center or in the Account Administration interface. If required, these Content-Security-Policy HTTP headers can be set in a reverse proxy to enable a Content Security Policy for the Manage and Signing Client.

**Manage Client (default CSP)**

Setting: `cirrus.rest.csp.value` (Administration Center only)

URI: `/cirrus`

```
default-src 'none'; style-src 'self' 'unsafe-inline'; script-src 'self'; img-src 'self'  
data: blob: https://signdoc-assistant.azurewebsites.net; frame-src 'self'; connect-  
src 'self'; font-src 'self'; media-src 'self'; object-src 'none'; form-action 'self';  
frame-ancestors 'self' https://teams.microsoft.com;
```

**Signing Client (with enabled Device Connector Support, default)**

Setting: `client.signing.csp.value` (can be account specific)

URI: `/cirrus/static/sc2`

```
default-src 'none'; style-src 'self' 'unsafe-inline'; script-src 'self'; img-src 'self'  
data: blob: https://signdoc-assistant.azurewebsites.net; frame-src 'self'; connect-  
src 'self' http://localhost:6613; font-src 'self' data:; media-src 'self'; object-src  
'none'; form-action 'self'; frame-ancestors 'self' https://teams.microsoft.com;
```

**Signing Client (with disabled Device Connector Support)**

Setting: `client.signing.csp.value` (can be account specific)

URI: `/cirrus/static/sc2`

```
default-src 'none'; style-src 'self' 'unsafe-inline'; script-src 'self'; img-src 'self'  
data: blob: https://signdoc-assistant.azurewebsites.net; frame-src 'self'; connect-  
src 'self'; font-src 'self' data:; media-src 'self'; object-src 'none'; form-action  
'self'; frame-ancestors 'self' https://teams.microsoft.com;
```

## Chapter 4

# Upgrade SignDoc Standard

SignDoc can be upgraded from any previous version. Unless specified in the version-specific sections below, the following generic upgrade procedure can be applied:

1. Stop SignDoc using `service_remove.cmd`.
2. Make a backup of the database or create a snapshot of the database that can be restored, in case the upgrade fails.
3. Install the new SignDoc version as described in [Quick start](#).
4. Apply the existing old SignDoc configuration to the new SignDoc installation. For example, apply any existing configuration from `application.properties` and `SignDocStandard.yml` (and possibly other configuration files) to the new installation. The configuration typically affects the database connection, SMTP server, `service.external.host.url`, and more.
5. Start the new SignDoc version using `service_up.cmd`.

**i** After the system is upgraded, it is no longer possible or supported to connect the old SignDoc installation to the upgraded/migrated database.

## Upgrade from SignDoc Standard 3.3.1 or earlier versions

The upgrade from SignDoc 3.3.1 requires some configuration changes. The following properties changed the name and must be adjusted for SignDoc 3.4.0 and newer versions.

The following table shows the old property name, the new property name, and the new environment variable name. The old property name can no longer be used.

| Old property name                      | New property name                       | New environment variable name           | Remarks                   |
|--|---|---|---------------------------|
| <code>jdbc.url</code>                  | <code>spring.datasource.url</code>      | <code>SPRING_DATASOURCE_URL</code>      |                           |
| <code>jdbc.username</code>             | <code>spring.datasource.username</code> | <code>SPRING_DATASOURCE_USERNAME</code> |                           |
| <code>jdbc.password</code>             | <code>spring.datasource.password</code> | <code>SPRING_DATASOURCE_PASSWORD</code> |                           |
| <code>jdbc.driverClassName</code>      | -                                       | -                                       | Obsolete                  |
| <code>SERVICE_EXTERNAL_HOST_URL</code> | <code>service.external.host.url</code>  | <code>SERVICE_EXTERNAL_HOST_URL</code>  | Used to be uppercase only |

| Old property name | New property name | New environment variable name | Remarks  |
|-------------------|-------------------|-------------------------------|--|
| SERVICE_HTTP_PORT | server.port       | SERVER_PORT                   | Obsolete and replaced by standard properties of spring |

Other changes are:

- The `CIRRUS_HOME` and/or `SIGNDOC_HOME` environmental variables are no longer used. The `signdoc_home` directory is now always located in the installation directory and cannot be located somewhere else.
- The `SignDocStandard.xml` file is replaced by the `SignDocStandard.yml` file. The `SignDocStandard.xml` file is no longer used.
- The `cirrus.properties` file is no longer used. The properties are now set in the `application.properties` file or in `SignDocStandard.yml`.
- The `service_configuration.properties` file is no longer used. The properties are now set in the `application.properties` file or in `SignDocStandard.yml`.
- It is generally recommended to use environment variables for configuration. The `SignDocStandard.yml` file is used to configure the SignDoc service on Windows and can be used to set environment variables. The `application.properties` file is used to set a base configuration. Configuration in `SignDocStandard.yml` will overwrite settings done in `application.properties`.
- Logging appenders are now configured in the `logback-spring.xml` file. The `logback-spring.xml` file is used to configure the logging output and destinations. The `logback-spring.xml` file is located in the installation directory.

## Upgrade from SignDoc Standard 2.2.1

If you are upgrading an existing SignDoc 2.2.1 installation (version  $\leq 2.2.1.2.0.63$ ) you are required to make configuration changes.

File: `service_configuration.properties`

- The `jdbc.password` configuration setting must not contain any of these 3 characters: `<` `>` `&`
- The `jdbc.url` configuration setting MUST NO LONGER be enclosed in quotes as it was true for SignDoc up to 2.2.1.2.0.63. Until now, the `jdbc.url` value had to be enclosed in single quotes, if there were spaces in `jdbc.url`. If this should be the case for the installation to upgrade, the single quotes must now be removed.

Example

Invalid settings:

```
jdbc.url='jdbc:h2:${SIGNDOC_HOME}/db/
signdoc_database;MVCC=TRUE;DB_CLOSE_DELAY=-1;INIT=SET COLLATION ENGLISH
STRENGTH SECONDARY'
```

**Valid settings:**

```
jdbc.url=jdbc:h2:${SIGNDOC_HOME}/db/  
signdoc_database;MVCC=TRUE;DB_CLOSE_DELAY=-1;INIT=SET COLLATION ENGLISH  
STRENGTH SECONDARY
```

## Upgrade troubleshooting

It should be sufficient to make sure that there are no PATH entries that point to an old SignDoc Standard installation. If an installation fails, verify the following:

- [General prerequisites](#) are met.
- The installation is performed on a supported Windows Server operating system specified in the [Tungsten SignDoc Technical Specifications](#) document.
- There should be no Windows PATH (system or user) entry (environment variable) pointing to an old SignDoc Standard installation.
- There should be no CIRRUS\_HOME environment variable set (system or user).


## Chapter 5

# Database migration

This chapter describes the database migration mechanism used by SignDoc Standard.

Any product that uses a schema-based database faces the challenge of managing database changes while migrating from one version to another. This includes changes to the database schema, such as adding a new column, moving data from one location to another, and more. Not only the database has to be adapted to the new schema, the existing data has to be migrated to fit it.

Database migrations standardize the way this is done, keeping track of the versions that have been applied to the data.

 We strongly recommend that you perform a database backup before attempting to migrate the database. Due to the nature of some migration steps and the fact that multiple migration steps are applied during a version update, a database rollback is not possible.

## Database migration with Flyway

SignDoc Standard uses Flyway to standardize database migration scripts. For more information on Flyway, see the documentation on the Flyway website. Flyway uses migrations that are named to a specific schema, containing the version number and description in the file name. It also keeps track of the version the database currently has and all applied changes by creating a database table named `schema_version` and recording all migrations it has done.

Since the migration scripts (or migration Java classes) are part of the product, you can always tell what state the database is in and what changes still need to be applied.

Flyway migrations are integrated in SignDoc Standard. When the application starts, it checks the database version and executes any outstanding migrations in the order of their version number, thus bringing the database up to date.

## Load balancing considerations

If multiple SignDoc nodes are hosted behind a load balancer (or reverse proxy), consider the following before upgrading to a new SignDoc version:

- Shut down all SignDoc nodes (instances) prior to starting the latest version.

- The first startup of the new SignDoc version should be done with a single node (instance) to process all migration tasks. After the first node is completed (the application or REST API is accessible), start the other nodes (instances).


## Chapter 6

# Troubleshooting

This chapter describes some issues you may encounter and their resolution.

## General

If you need to contact Tungsten Automation Support, it is useful to collect the following log and configuration files.

 Make sure to remove sensitive information.

Log files

- **logs directory:** Contains the log files of the SignDoc Standard application.
- **application.properties:** Contains the configuration of the SignDoc Standard application.
- **SignDocStandard.yml:** Contains the configuration of the SignDoc Standard service.
- **logback-spring.xml:** Contains the logging configuration of the SignDoc Standard application.
- **Browser console log:** Contains hints to client side issues of the SignDoc Standard application.

## Sanity Check

Make sure to complete this checklist, to spot potential issues early.

- Open `http://<server>/cirrus/test/cookie` and check that the test result is passed. See also [Cookies](#).
- Log in to the Administration Center and check all warnings or errors listed in **System information**.
- If you have already created a SignDoc Account, log in as Account Administrator and check all warnings or errors listed in **Status information**.

## SignDoc Standard does not start

Make sure that the database connection is configured properly. When the database connection is not configured properly, SignDoc Standard will not start. Check the log files for more information.



## SignDoc Standard does not send emails

Make sure that the SMTP server is configured properly. When the SMTP server is not configured properly, SignDoc Standard will not send emails. Check the log files for more information.

A default email configuration can be done via environment variables, `application.properties` or entirely in the Administration Center. At the same time, every SignDoc account may have its own SMTP configuration and overwrite the defaults. This can potentially cause unexpected issues if the default configuration uses settings that are in conflict with account-specific settings that the account does not overwrite explicitly.

Thus, when using multiple accounts in SignDoc that potentially use a dedicated SMTP server, it might be advisable to not set any default SMTP configuration but to configure it per account.

## Appendix A

# Glossary

### A

#### **Account**

A SignDoc account is a logical entity that contains all data and configuration for a specific customer or tenant. An account can have multiple users and multiple documents.

### J

#### **JDBC**

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database.

#### **JMX**

Java Management Extensions (JMX) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (such as printers), and service-oriented networks.

### R

#### **REST**

Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating web services.

### S

#### **SMTP**

Simple Mail Transfer Protocol (SMTP) is an Internet standard for email transmission.

#### **SSL**

Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client, typically a web server (website) and a browser, or a mail server and a mail client.

#### **SSO**

Single Sign-On (SSO) is a property of access control of multiple related, but independent software systems. With this property, a user logs in with a single ID and password to gain access to a connected system or systems without using different usernames or passwords, or in some configurations seamlessly sign on at each system.

### T

**TLS**

Transport Layer Security (TLS) is a cryptographic protocol that provides secure communication over a computer network. The protocol is widely used for secure web browsing and email.

**U**

**User**

A SignDoc user is a person who can log in to SignDoc and prepare and send out documents. A user can be a member of one account.