KOFAX

Kofax SignDoc®
E-sign with Kofax

Kofax SignDoc 1.3.1

Installation Guide

Table of Contents

1 Introduction

Kofax SignDoc transforms customer experiences by streamlining the signing of documents. SignDoc accelerates business workflows by removing steps such as printing, routing, and shipping documents back and forth. Constituents can sign electronically on any device anywhere resulting in significant operational cost reductions, productivity increases, and improved compliance.

Kofax SignDoc

- is designed to manage document-based transactions in a multi-channel environment.
- is the fastest way to get a signature from people in a convenient and secure process.
- is an end-to-end solution enabling preparation, execution and management of transactions in a digital environment across organizations and individuals.
- is compliant with federal e-signature legislation, which gives electronic documents and signatures the same legal standing as paper documents and ink signatures.
- provides a signed document containing all audit information if desired (self-contained document).
- supports multiple signature types such as click-to-sign, handwritten, and photo.
- can be operated behind the firewall (in-house) or in an enterprise cloud environment.

The Kofax SignDoc fundamentals are

- Intuitive UI (self manageable process)
- Customizable workflow (who, what, where, how & when, expiration, call back)
- Signer authentication (options and API)
- Sign anywhere and anytime – individually adapted to environment (click, mouse, photo, handwritten - be mobile and flexible)
- Audit trail included in signed documents
- Self-contained documents ("standard" PDF viewer also for audit trail)

For enterprise workflow applications Kofax SignDoc provides integration interfaces via web services.

Kofax SignDoc has been designed to be flexible - for Kofax SignDoc users and recipients of SigningPackages.

NOTE

Recipients of Kofax SignDoc do not need a Kofax SignDoc account.

1.1 Target Audience

The installation guide is intended for those who want to install Kofax SignDoc and its components.

1.2 Kofax SignDoc - General Overview

Layer 1 - Operating Systems

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

Layer 2 - Kofax SignDoc Application

The application consists of two WARs (Web application Archive) which are deployed into a Web Application Server. The **sdweb.war** file contains Native Libraries which are used for licensing and PDF handling. The configuration files of **cirrus.war** and **sdweb.war** are stored outside of the WAR files in folders referenced by environment variables **CIRRUS_HOME** and **SDWEB_HOME**.

Layer 3 - REST Interface

It is possible to interact with the system via REST API that supports almost all aspects of the application. Amongst many other things, it is possible to create and schedule Signing Packages with one REST request. A detailed API documentation is included.

Layer 4 - Web Application Server

Kofax SignDoc runs on the standard Web Application Servers used within the industry: Apache Tomcat. For supported versions and prerequisites please check the *Kofax SignDoc Technical Specifications* document.

2 Standard Installation

General Notes

SignDoc Standard can be run on Windows and Linux operating systems. For a list of supported environments please have a look at the *Technical Specification Document*. This guide assists in setting up a standard installation on a Windows 64 bit. Screenshots might look different depending on the Windows Version used. For an installation on Linux systems, please have a look at the provided and documented Dockerfile.

The installed system consists at least of these components:

- Database (MS-SQL Server)
- Application Sever (Apache Tomcat + SignDoc Standard)

SignDoc Standard is executed as J2EE compatible application in the application server. Apache Tomcat and SQL Server are usually installed on different computers (nodes) for various reasons, but they can also be installed on the same instance. This installation guide references and distinguishes these nodes by the labels **sql-node** and **app-node**. Both labels act also as the instance's DSN name.

Remark: If the Application Server and Database are installed on the same OS instance...

- **sql-node** can be substituted with ***localhost***.
- **app-node** can also be substituted with ***localhost***, if the system is only accessed locally, what may be useful for very simple test or demonstration purposes.

Reverse proxy / Load Balancing

This installation guide does not consider/discuss the setup up of a Reverse Proxy or Load Balancing. Nevertheless, it is important to note that a Reverse Proxy / Load Balancing setup must respect sticky sessions cookie (usually JSESSIONID).

SSL Setup

This installation guide does not consider/discuss a SSL configuration, since this usually depends on local IT regulations and is effectively transparent to SignDoc Standard.

Software requirements for this guide

Please check the *Kofax SignDoc Technical Specifications* document for version details.

- Java 8 Runtime Environment
- Windows SQL Server (the express version is sufficient)
- Apache Tomcat

2.1 Windows 64 Bit

Installation on Windows 64 Bit consists of these basic steps:

- Database Installation and Configuration
- Java Installation
- Apache Tomcat Installation and Configuration
- SignDoc Standard Installation and Configuration

2.2 Install and Configure Microsoft SQL Server

For production purposes, SignDoc Standard requires a database server to be able to store application data. Currently Microsoft SQL server is supported. While installing the database server, use the suggested defaults unless noted otherwise.

For this guide we will use Microsoft SQL Server 2012 as database service.

REMARK

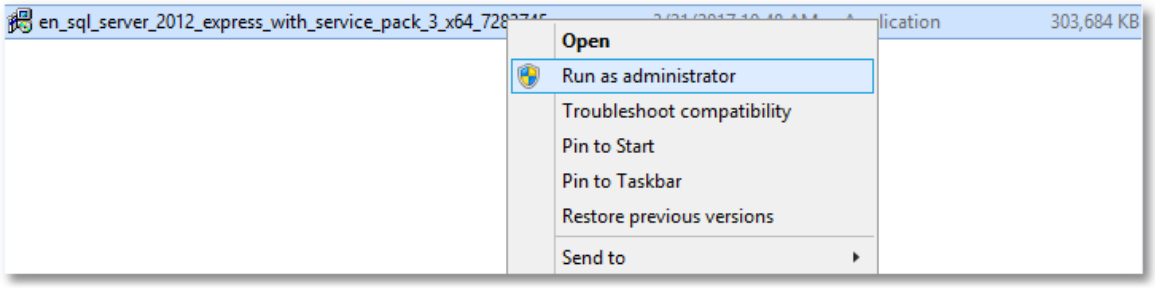
For the DNS name of the database instance we will use the convention **sql-node**. This must be substituted with the correct DNS name of the database server

2.2.1 Database Installation

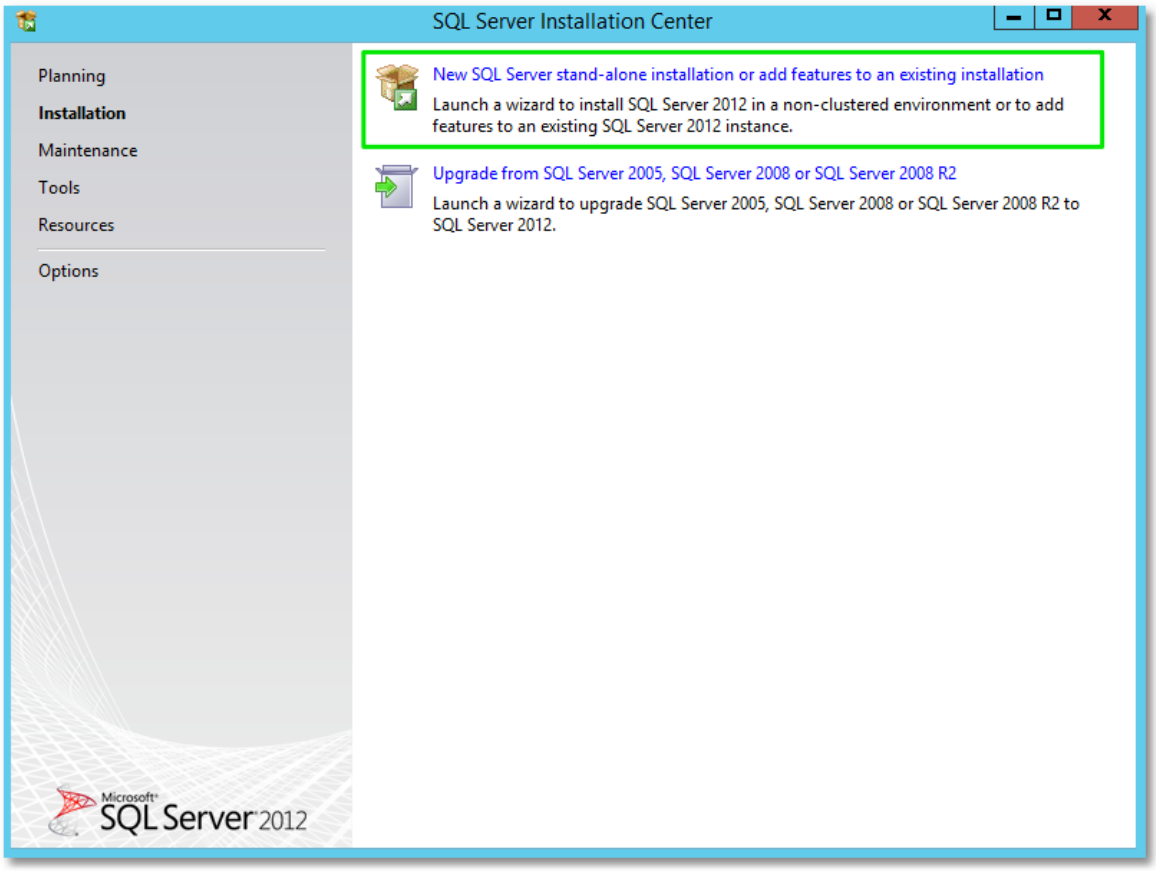
Example

MICROSOFT SQL SERVER 2012 EXPRESS
(en_sql_server_2012_express_with_service_pack_3_x64_7283745.exe)

Install with administrator rights (if required).

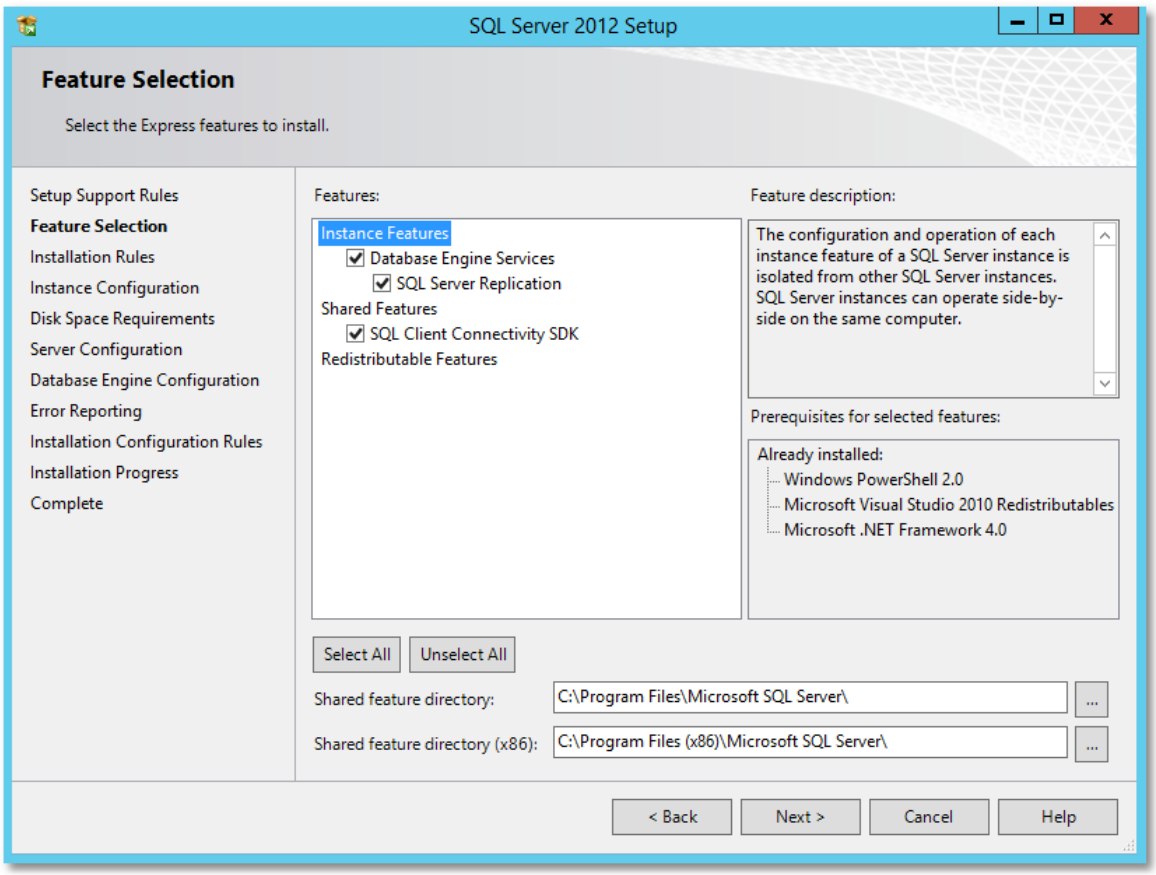


Select "New SQL Server stand-alone installation or add features to an existing installation":

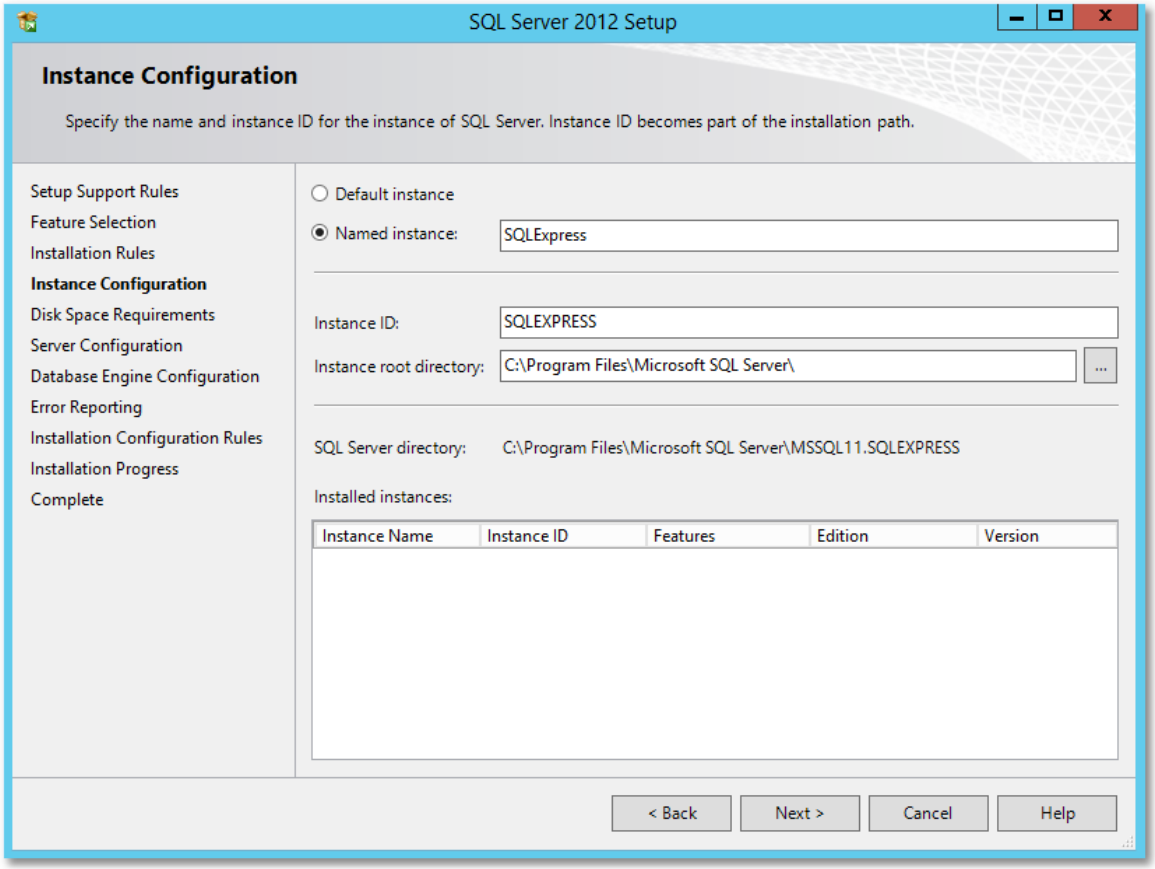


Feature Selection

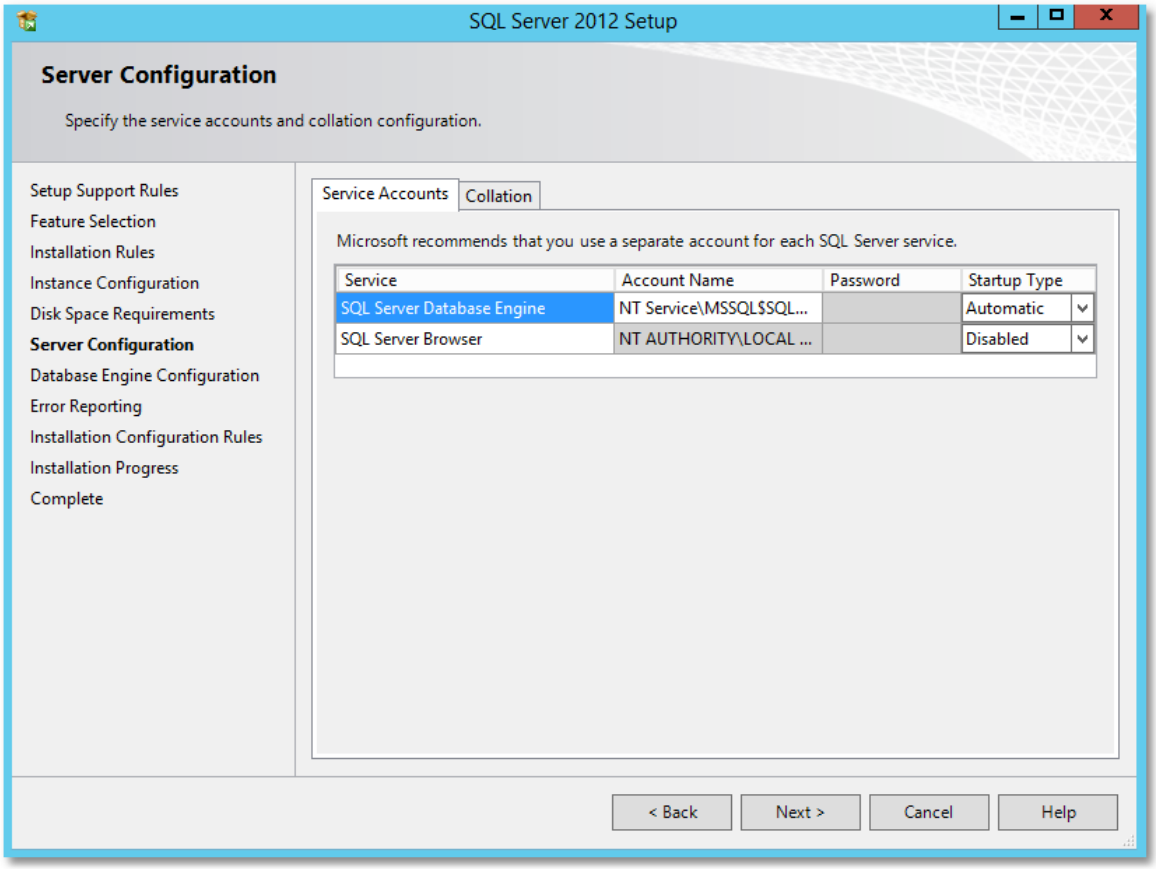
Select all features:

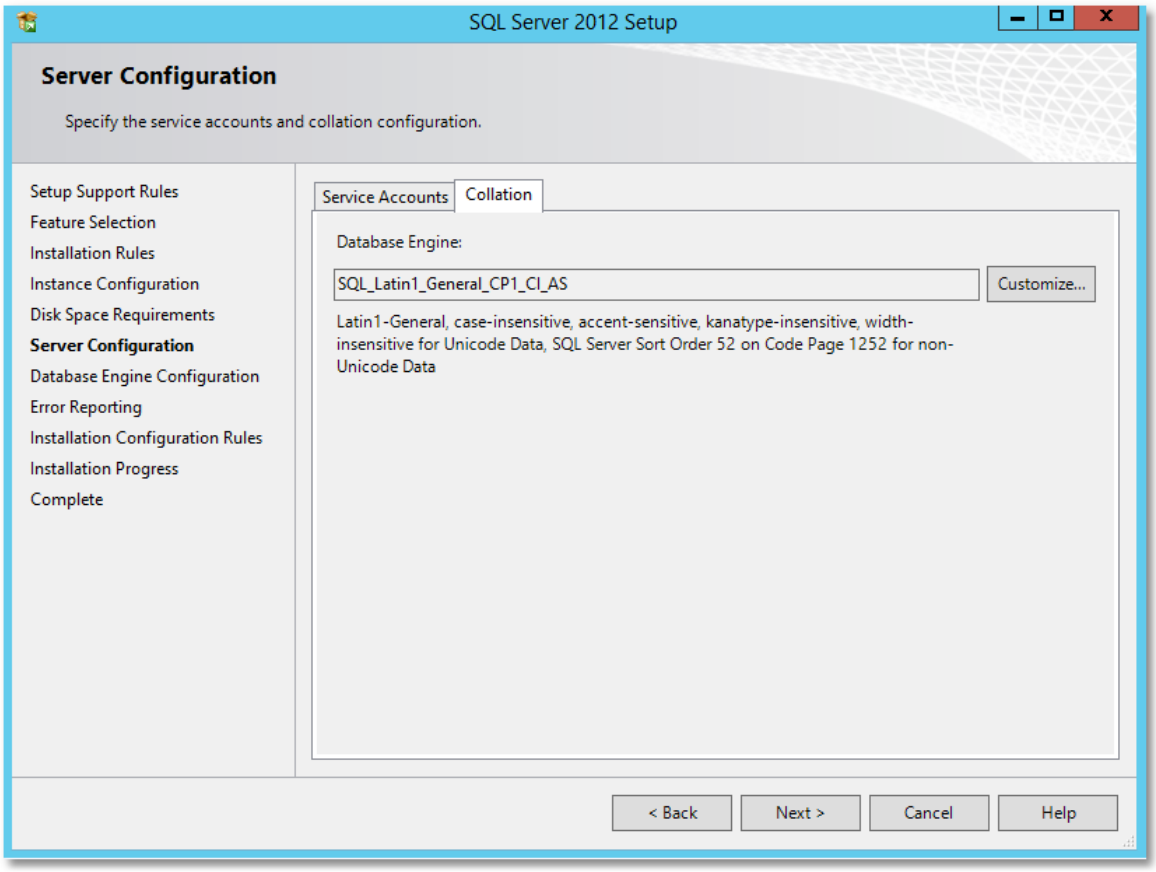


Instance Configuration



Server Configuration



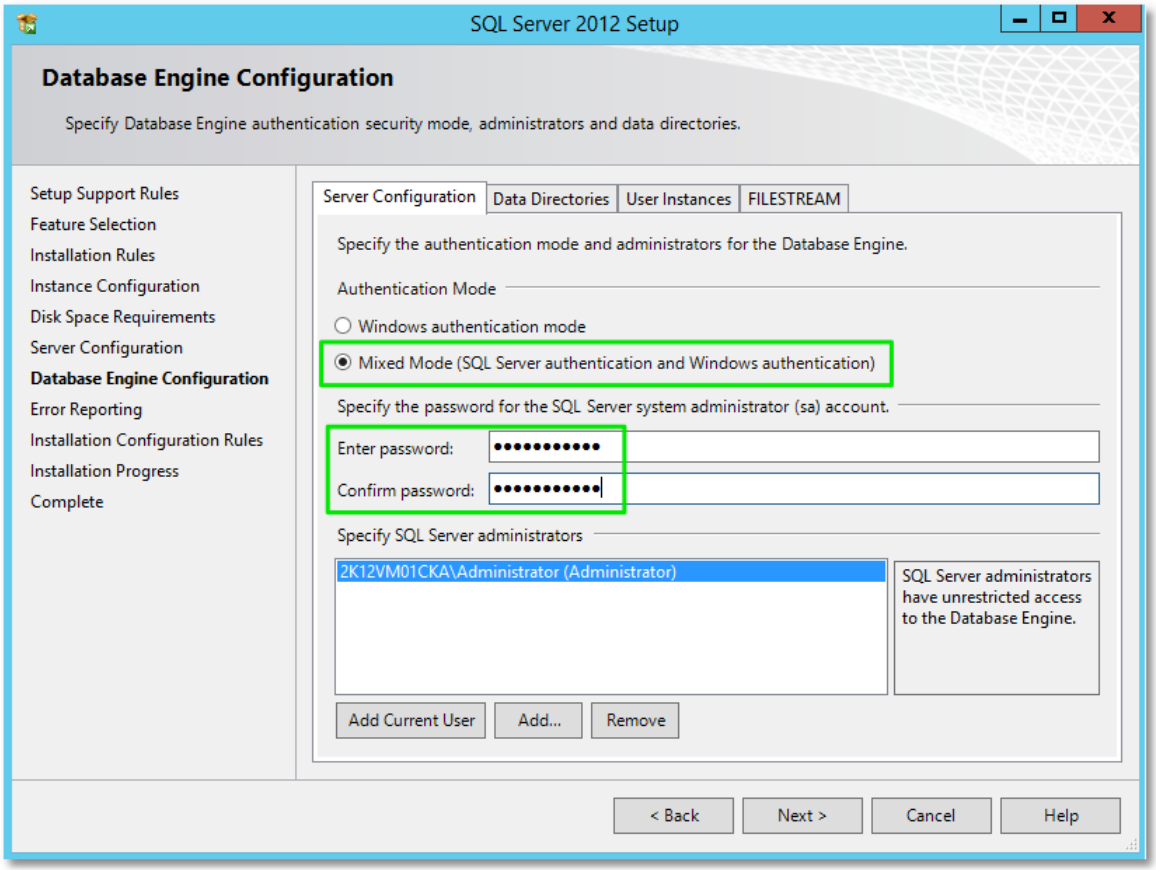


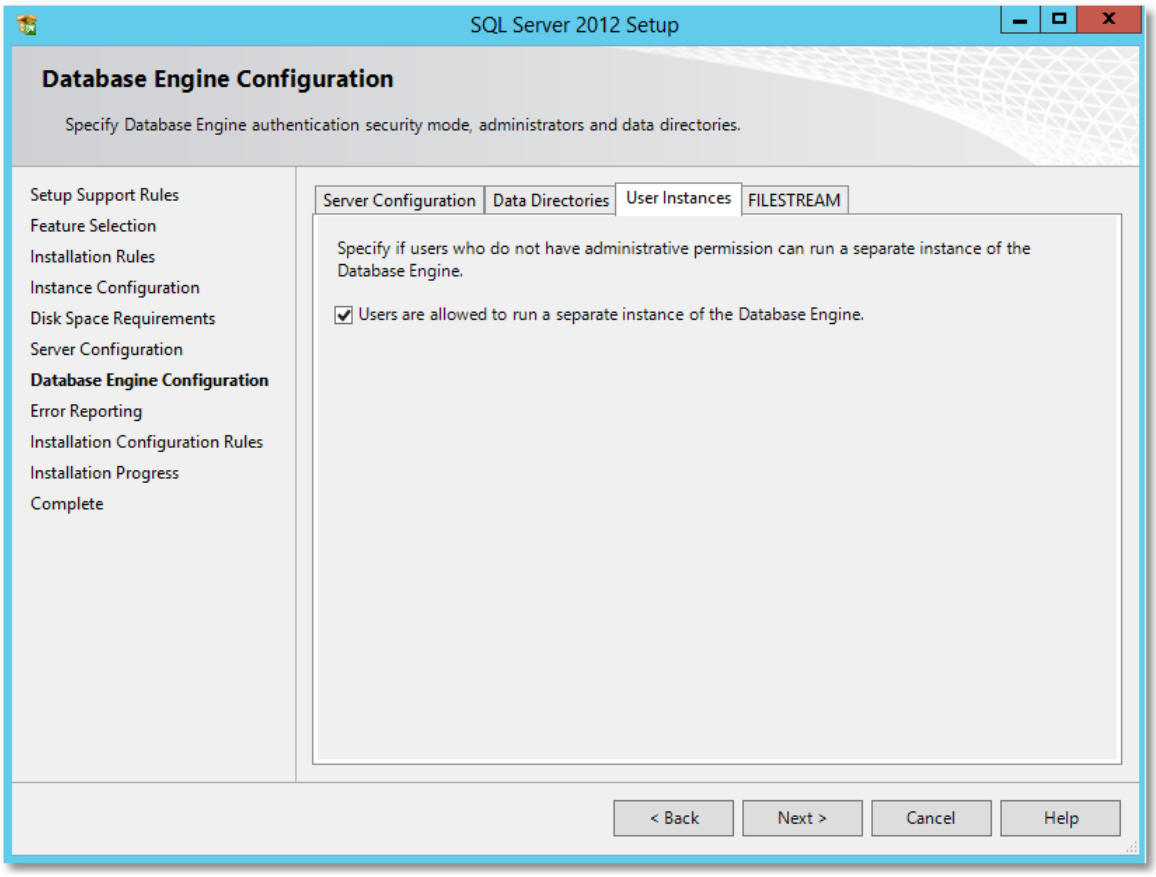
2.2.2 Database Engine Configuration

Example

MICROSOFT SQL SERVER 2012 EXPRESS
(en_sql_server_2012_express_with_service_pack_3_x64_7283745.exe)

Select **Mixed Mode** and define the password for the system administrator (sa) user. We will use **2beChanged!** for this guide.





Finish the installation.

Make sure that the server listens on the desired TCP port for connections. For this guide we will use port 1433.

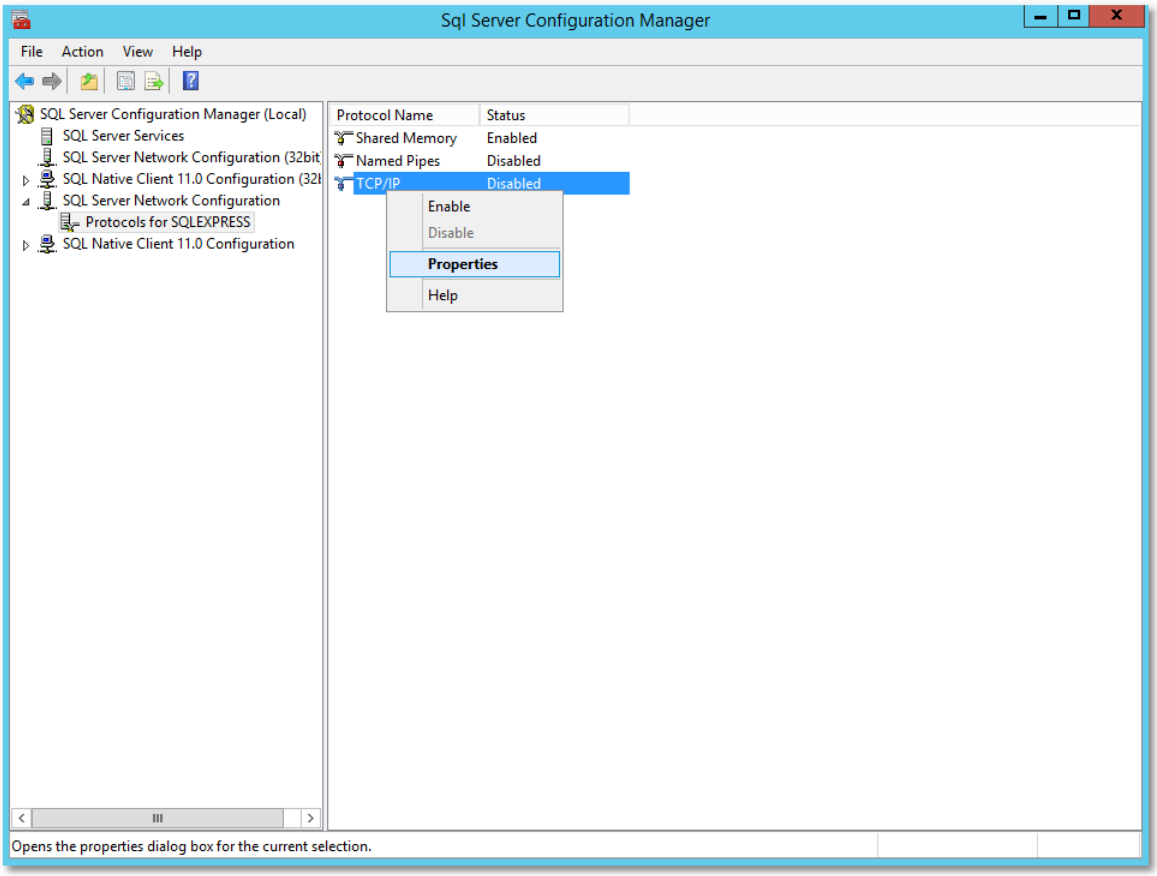
Example

Start

(run as administrator) and select:

SQL Server Network Configuration > Protocols for SQLEXPRESS

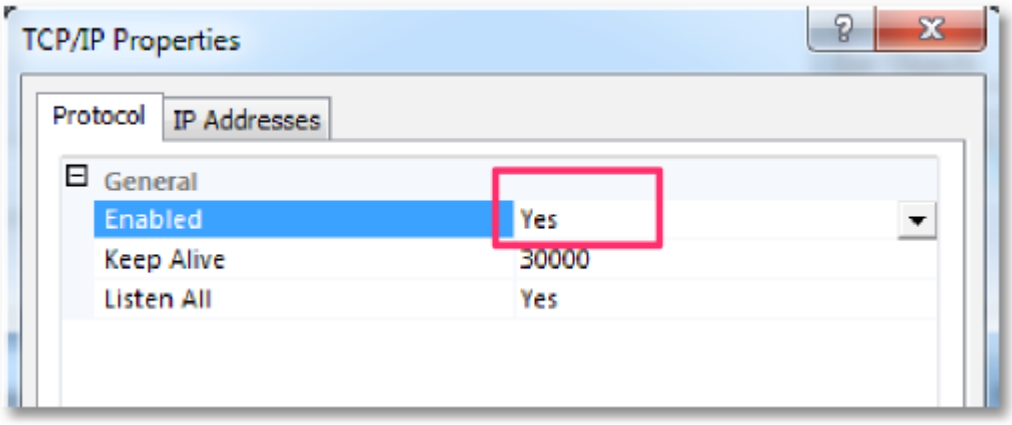
Right-click on **TCP/IP** to select **Properties**:



Select tab **Protocol** and set property:

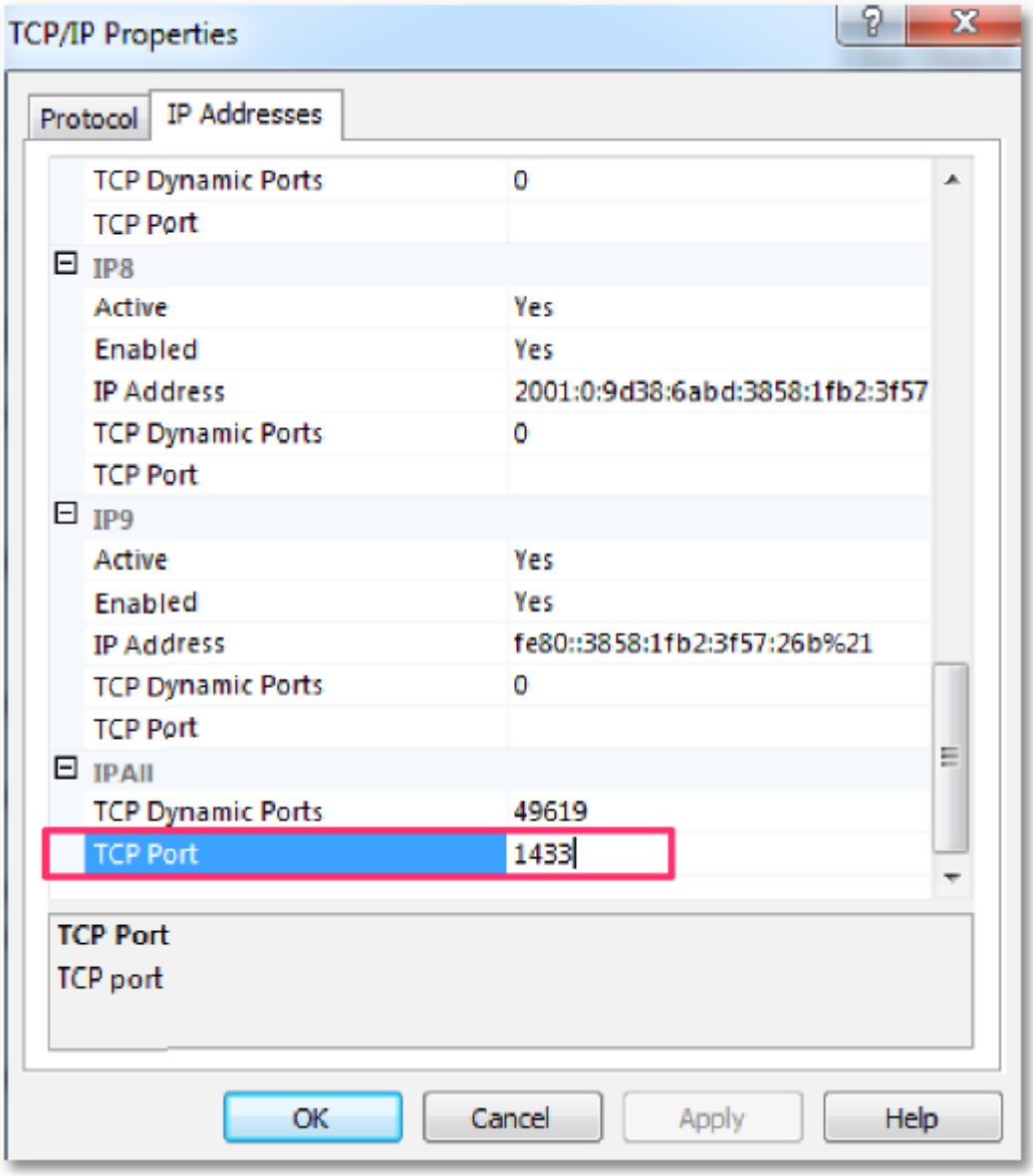
Enabled = Yes

Apply changes with **OK**.



Select tab **IP Addresses**, scroll down to section IPALL and enter value for TCP Port:

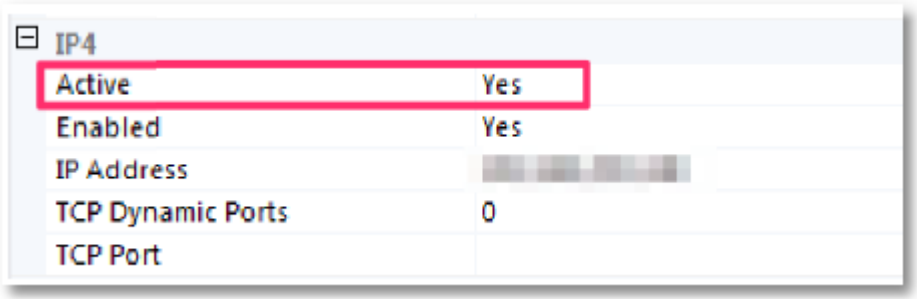
PCP Port = 1433



Make sure to activate the IP Addresses the server should listen on:

Active = Yes

It's possible to activate all first and limit it later to the real needs.



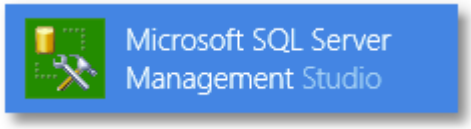
Apply changes with **OK**.

Stop and start service.

Create an empty or new database instance. For this guide we will use **signdoc**.
A new database instance can be created using the SQL Server Management Studio (SSMS).

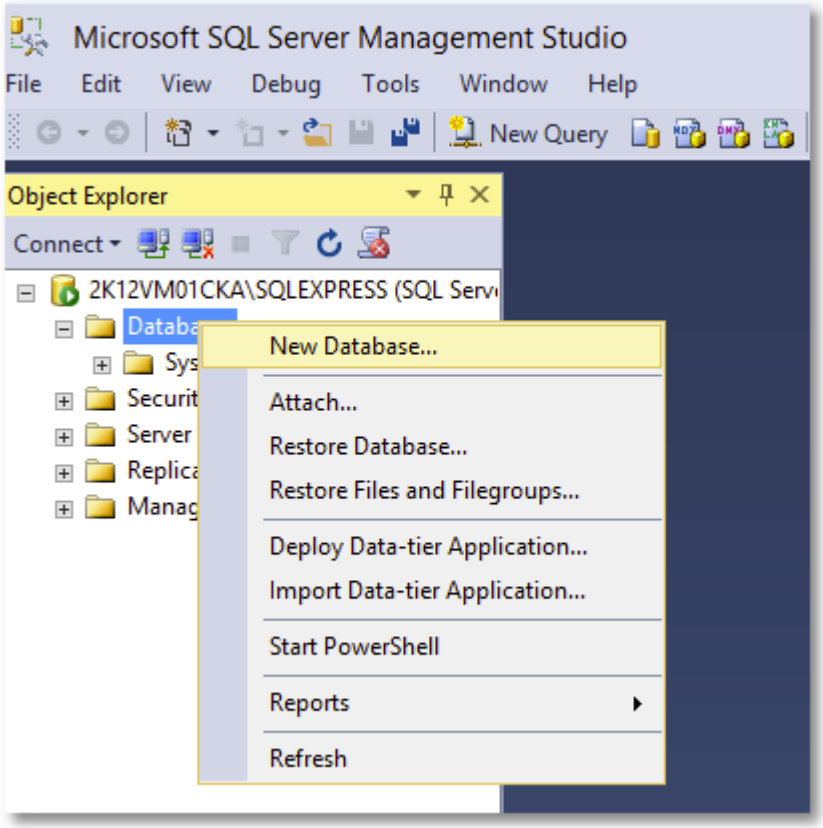
Example

Download and Install SSMS from Microsoft with administrator rights.
Start SSMS

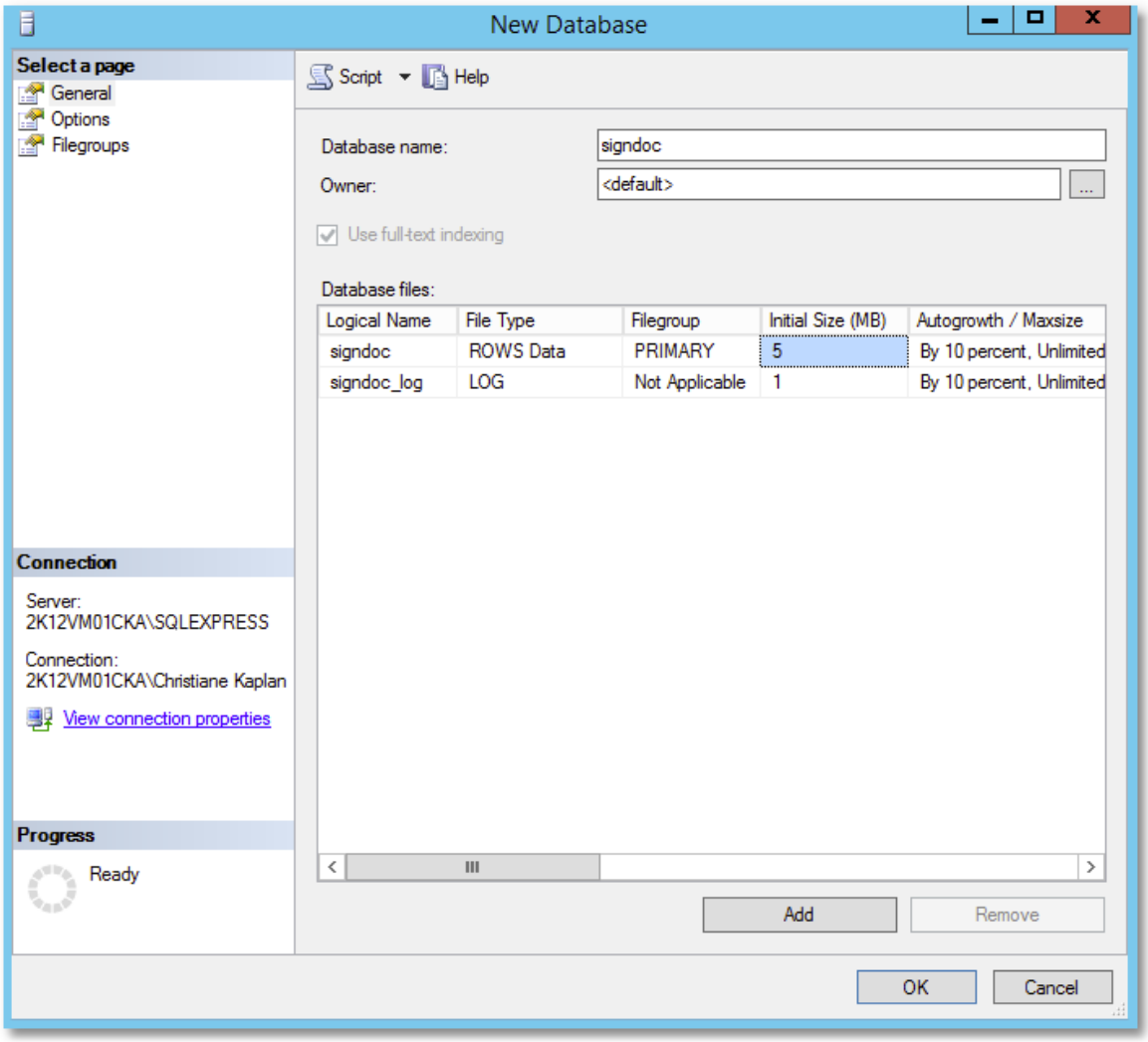


Connect to the Server.

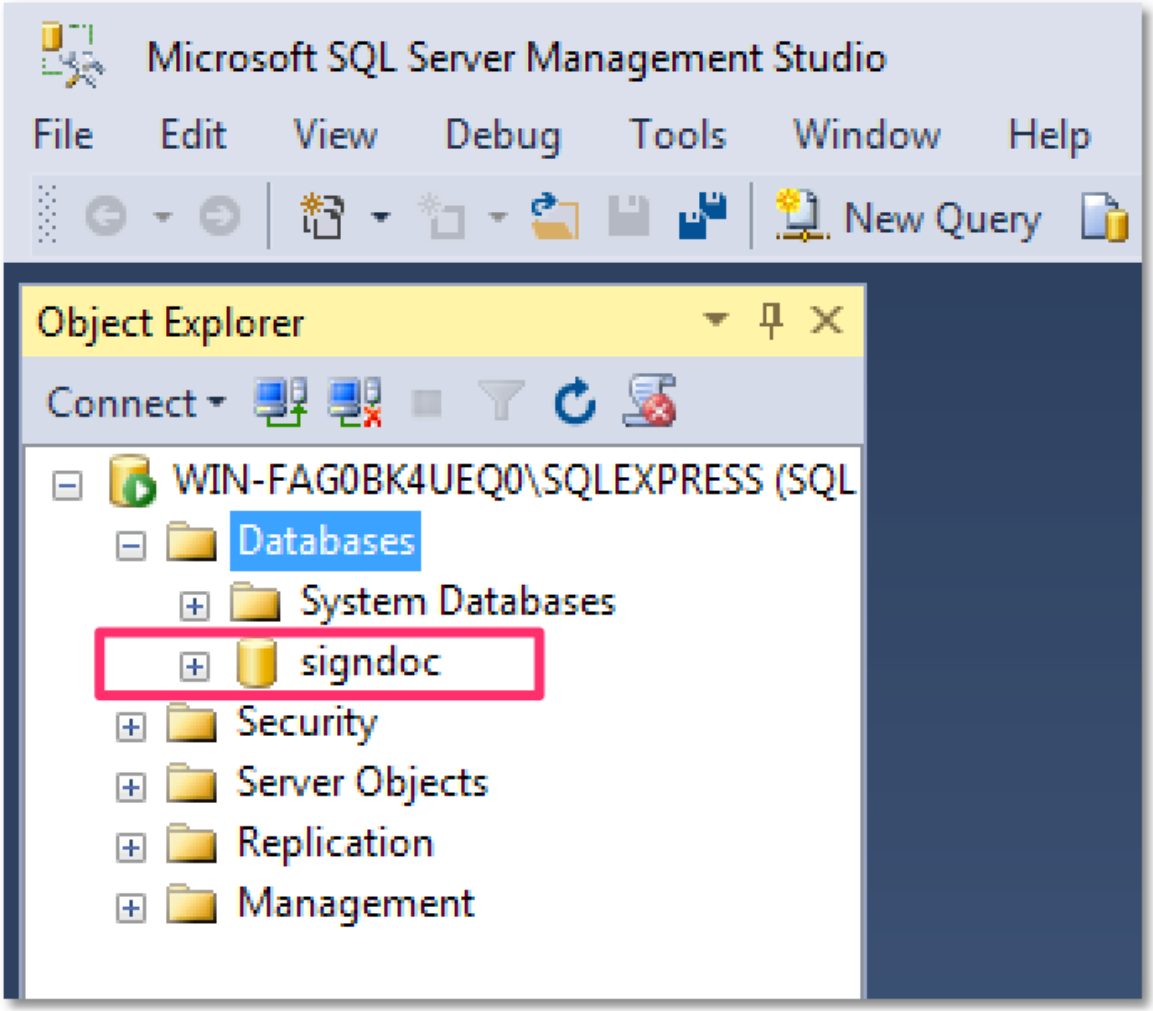
Create a new database. Right click on **Database** and select **New Database**.



Enter a name for the database. For this guide we use **signdoc** as database name.



New Database



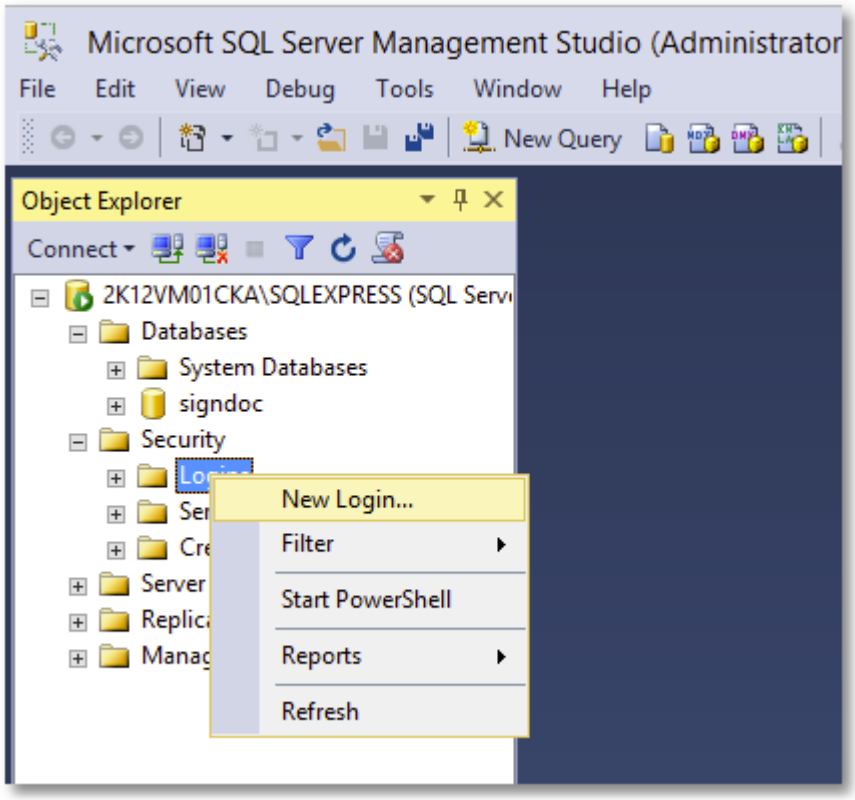
Create a new database user that is a member of the role **db_owner** of the **signdoc** database and uses "SQL Server authentication".

For this guide we will use **signdoc** also as user name. The new user requires also a password – make sure to unselect “”.

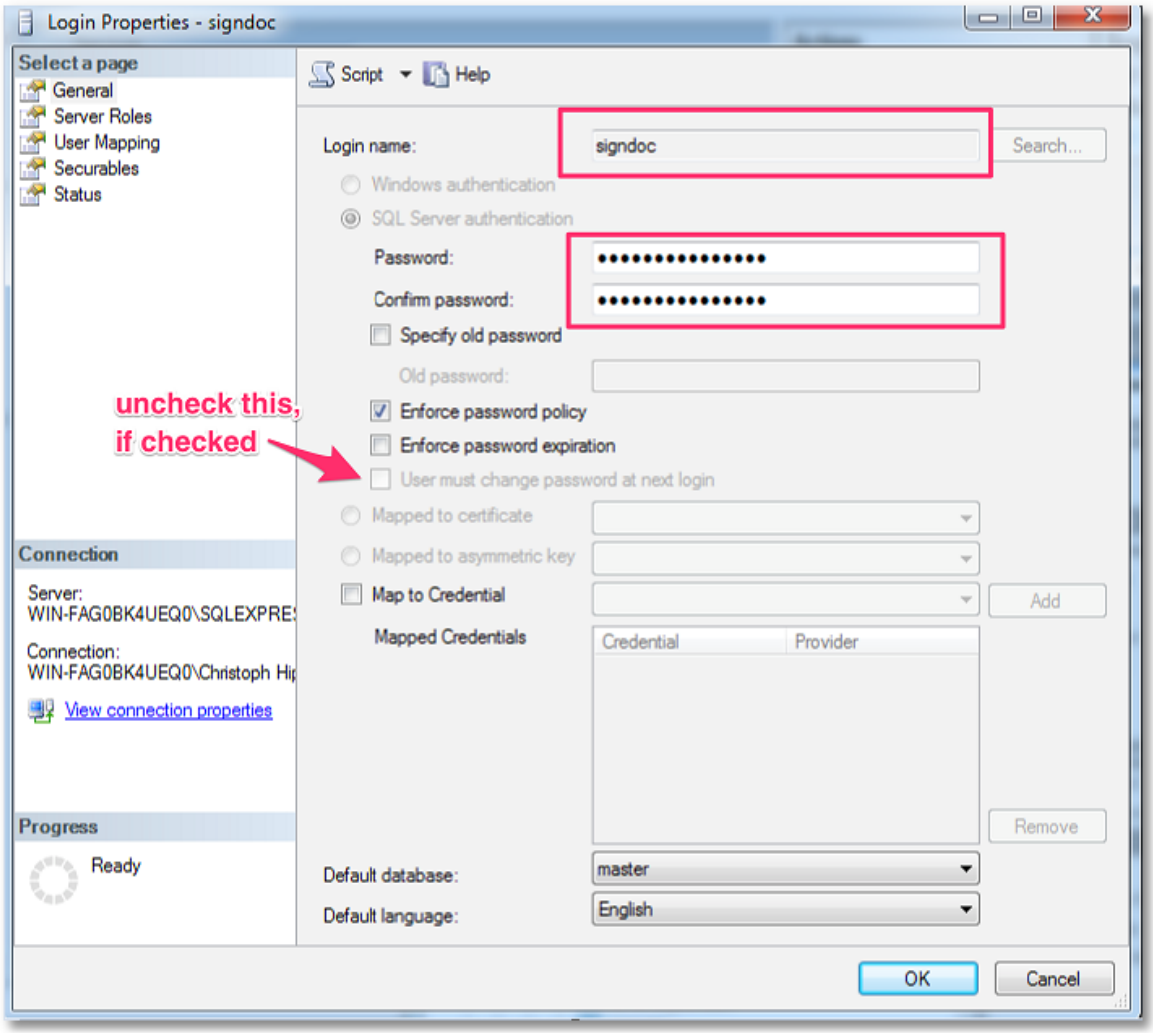
For this guide we will use:

2beChanged!

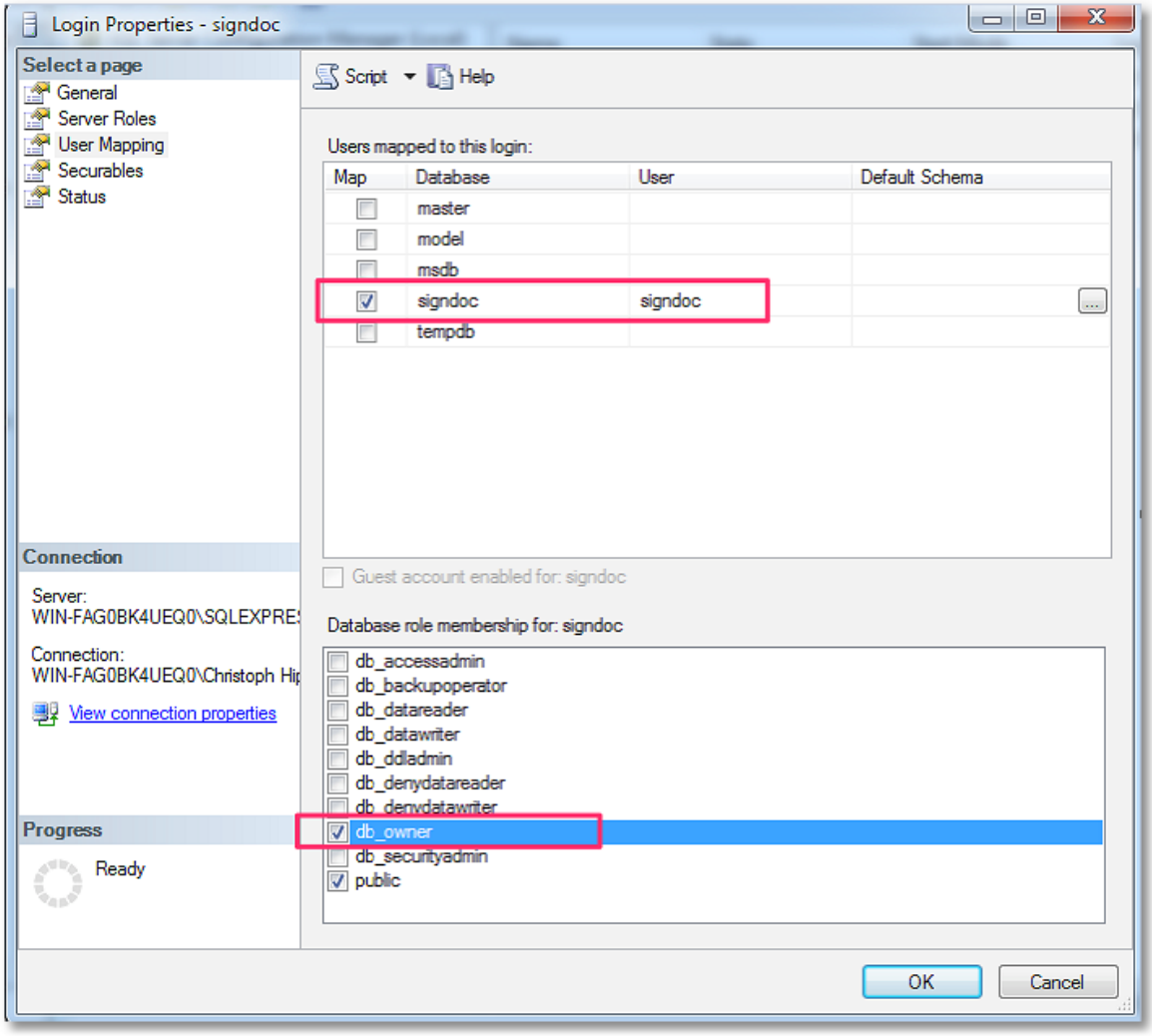
Create a new login. Open **Security** and right-click on **Logins**.



Go to **General Pane** and enter the according information.



User Mapping pane and select database **signdoc** and role membership:



2.3 Install and Configure Java 8 Runtime

SignDoc Standard requires a Java 8 Runtime (64 Bit) with updated Java Cryptography extensions to run.

Remark: For the DNS name of the Java instance we'll use the convention **app-node**. This must be substituted with the correct DNS name of the server where the Java 8 Runtime (64 Bit) is installed

2.3.1 Install the Java Runtime 8

Download and install the latest **Server JRE (Java SE Runtime Environment) 8** from Oracle:

Navigate to Java SE downloads
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Open the latest Server JRE (Java SE Runtime Environment) 8 Downloads page.

Download and install the **Windows x64** version.

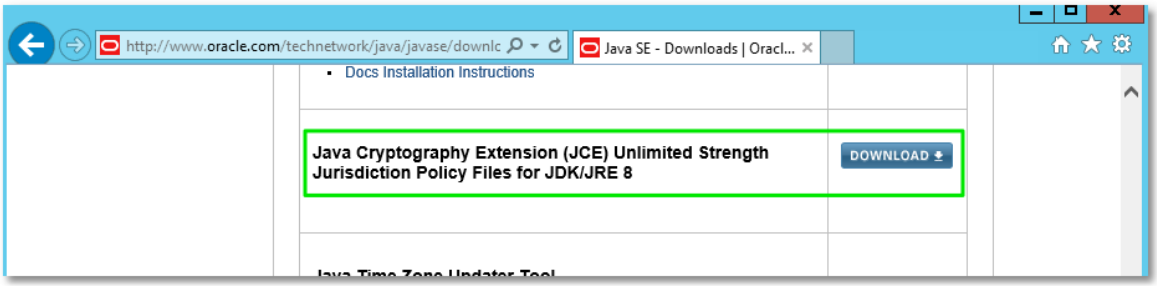
Example: The current version is: 8u121 (at the time of the writing of this document)
http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html

2.3.2 Install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction

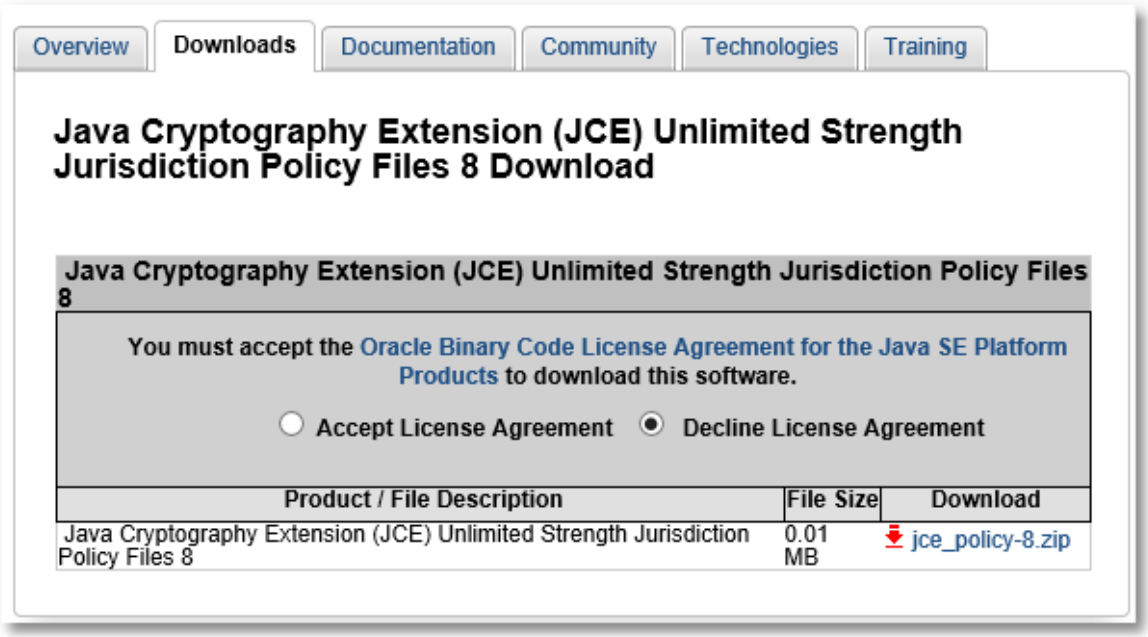
Download and install from Oracle:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Look for Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8.



Select **Download**:

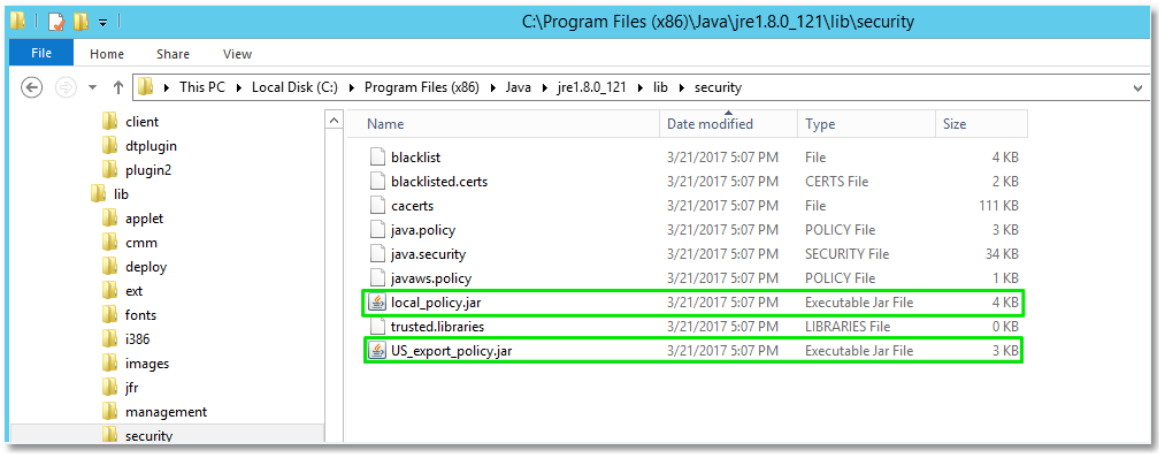


Open the Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX.

Be sure that:

- Your browser has "cookies" and JavaScript enabled.
- You clicked on "Accept License" for the product you wish to download.
- You attempt the download within 30 minutes of accepting the license.

After download unpack the zip file and follow the installation instructions described in the readme.



2.4 Install and Configure Apache Tomcat 8.x

SignDoc Standard requires a Apache Tomcat 8.x application server to run.

REMARK

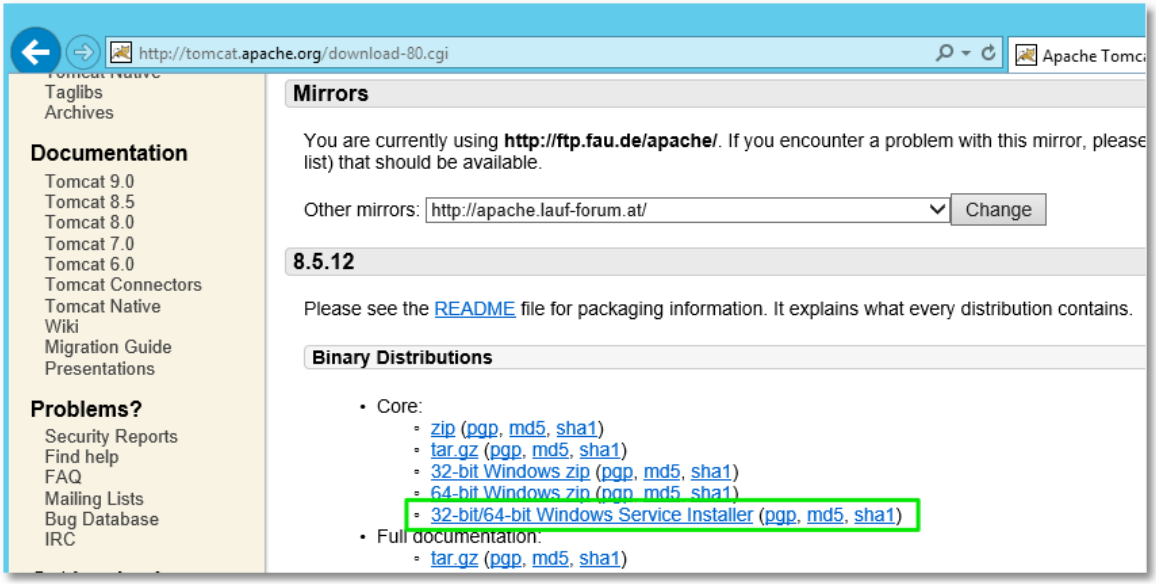
For the DNS name of the Apache Tomcat 8.x instance we'll use the convention **app-node**. This must be substituted with the correct DNS name of the server where Apache Tomcat 8.x is installed. This is also the same instance where Java was installed in the previous chapter Install and configure Java 8 Runtime.

2.4.1 Download Installer for Apache Tomcat

Download Installer from Apache:

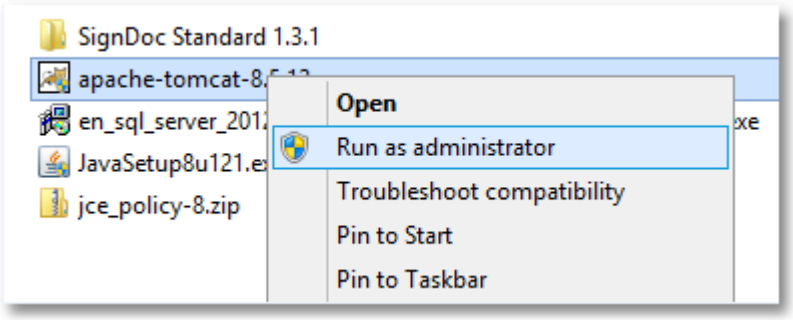
http://tomcat.apache.org/download-80.cgi

Use the 32-bit/64-bit Windows Service Installer:



Example

apache-tomcat-8.5.12.exe

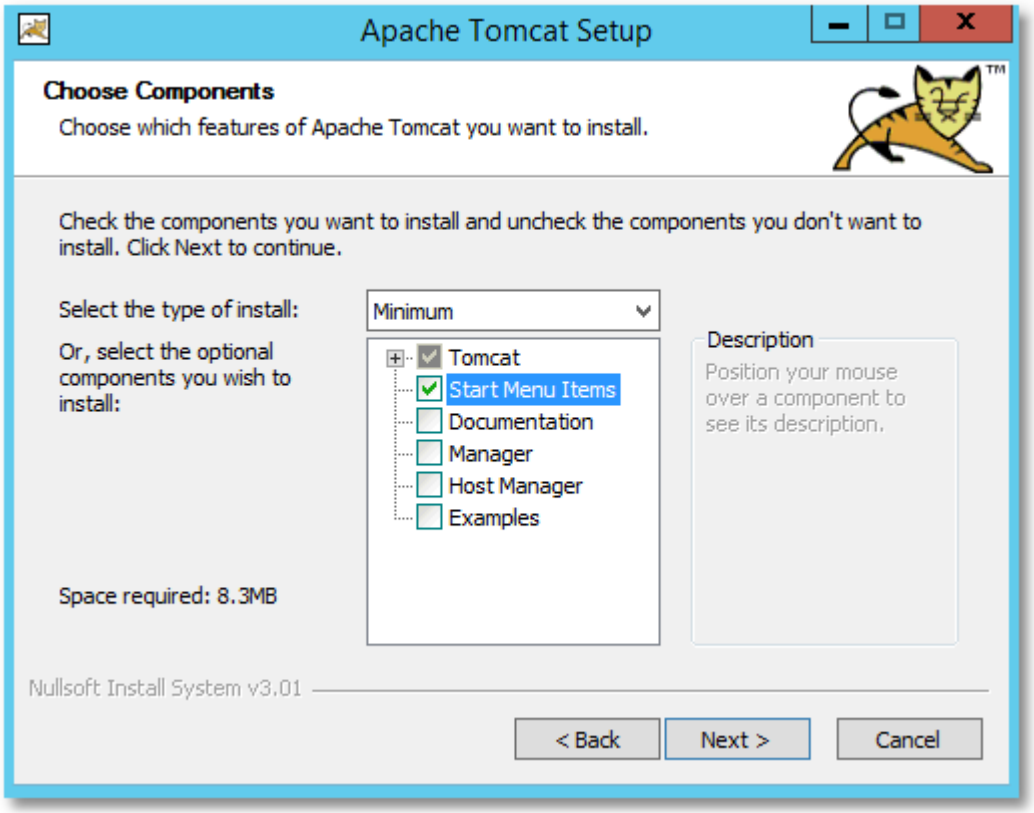


2.4.2 Install Apache Tomcat Application

Use the defaults unless noted otherwise.

Choose Components

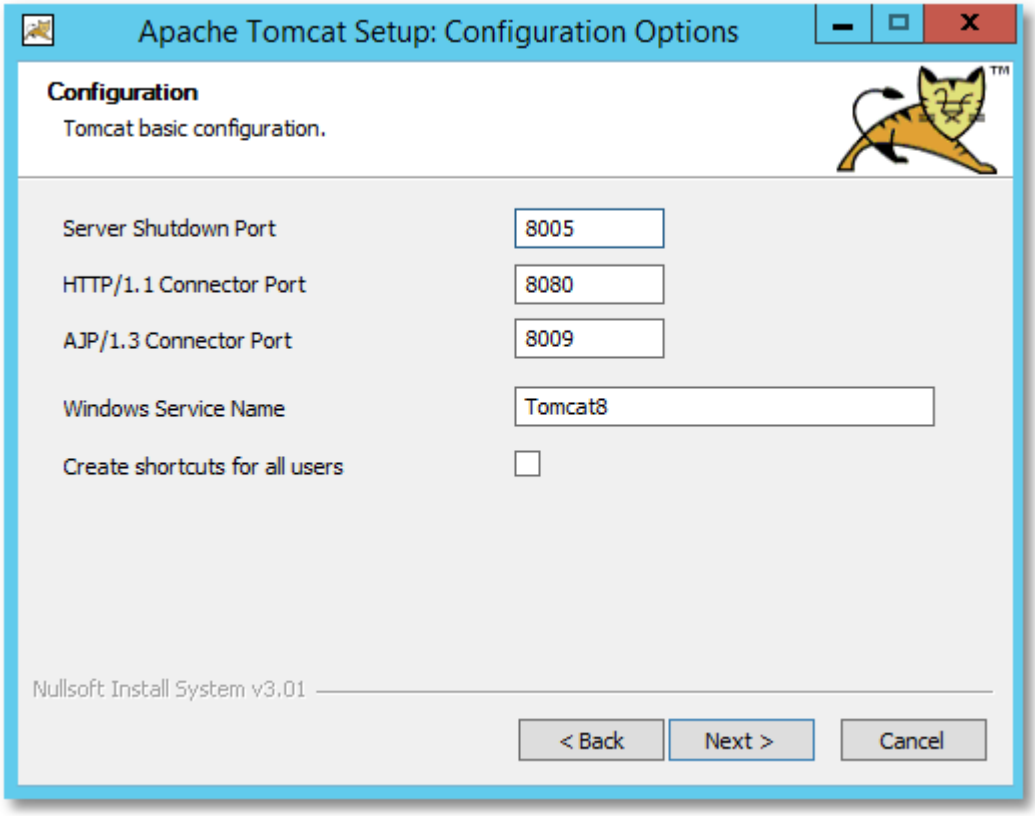
The options **Documentation** and **Manager** can be deselected:



Configuration

Adjust the values to your needs. It's fine to use the defaults, as long as there is no other local service is using the ports. If unsure, ask your system administrator.

Example



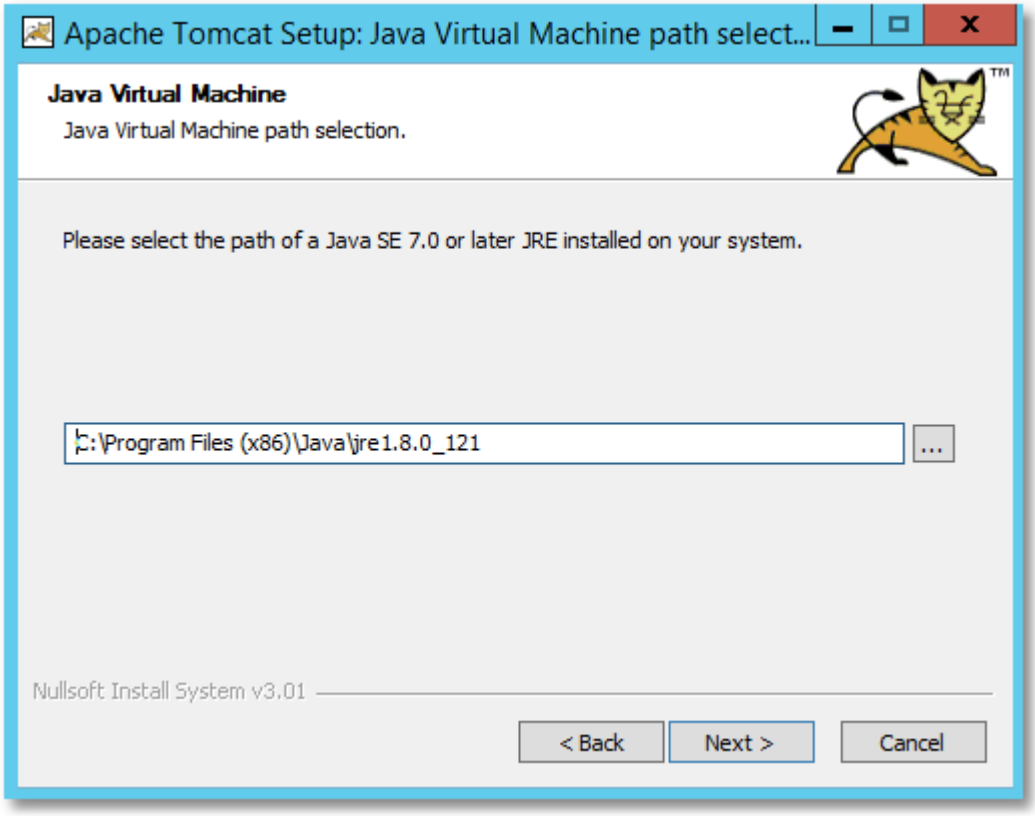
Java Virtual Machine

The previously installed Java Runtime should be detected automatically. If not, choose the correct path to the Java 8 Installation of Install the Java Runtime 8.

IMPORTANT NOTE

This directory will be referenced as ***%CATALINA_HOME%***

Example



Set important Java options

The Java options **Java Heap** and **-Dfile.encoding=UTF-8** must be set:

Java Heap

This defines the maximum heap size that Java allows for the application. This should be set at least to 2 GB.

For this guide we will use: 2GB (2048 MB) as max value and set it in the “Configure Tomcat” tool.

This can be manually set via standard Java options: ***-DXmx2G***

NOTE

If the machine will not be able to reserve this space Tomcat will not start. The tomcat logfile will display a message similar to:

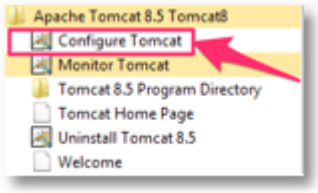
> Error occurred during initialization of VM Could not reserve enough space for 2097152KB object heap.

If so, please reduce your heap to size fitting to your machine.

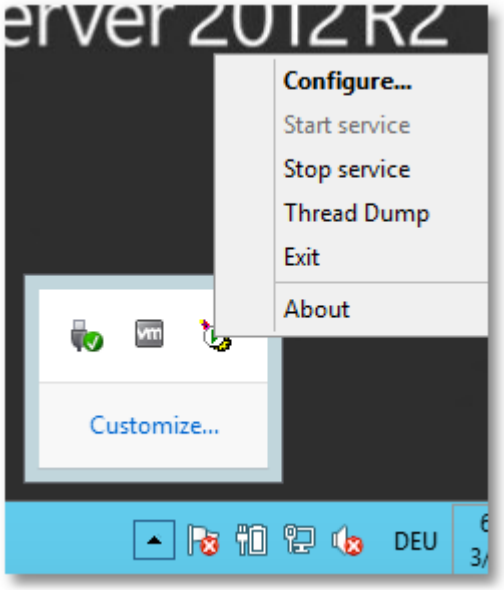
-Dfile.encoding=UTF-8

This option is required so SignDoc Standard handles non-ASCII characters properly.

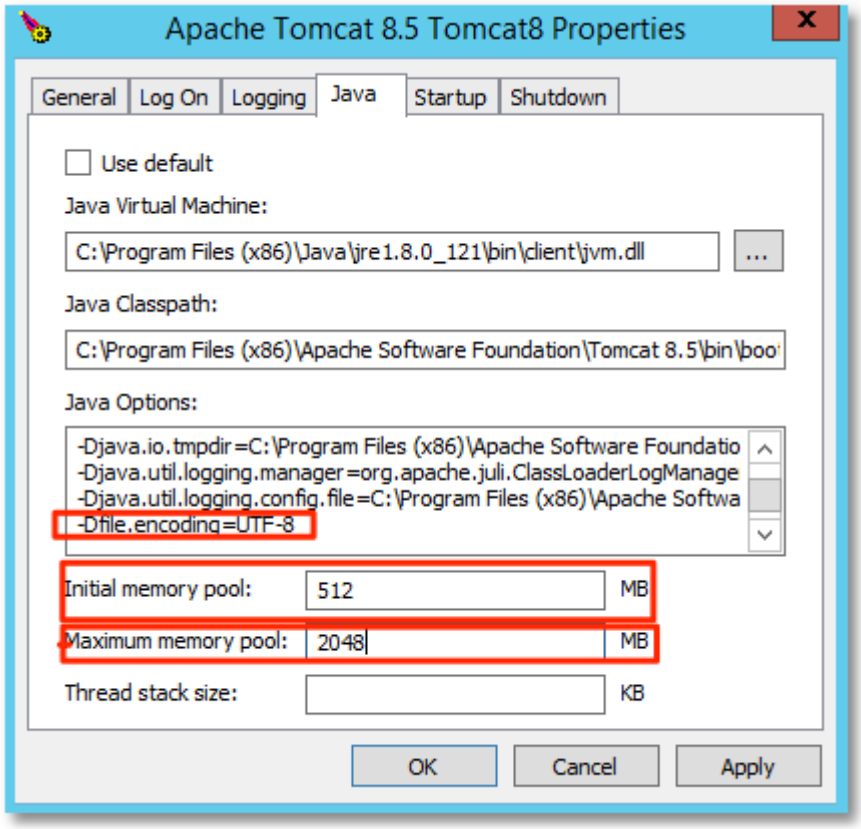
Example


Apache Tomcat 8.5 Tomcat8
Configure Tomcat
Monitor Tomcat
Tomcat 8.5 Program Directory
Tomcat Home Page
Uninstall Tomcat 8.5
Welcome

or


Configure...
Start service
Stop service
Thread Dump
Exit
About
Customize...
DEU

Go to tab **Java** and set and apply the options.

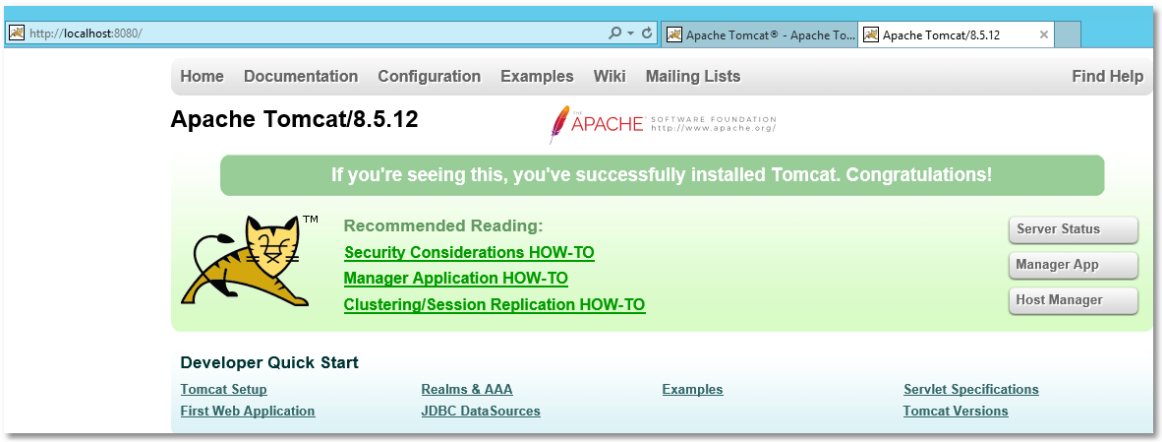


Restart the Tomcat Service.

Verify Tomcat installation

After the installer is completely finished, open http://localhost:8080/ in a web browser. The Apache Tomcat start page should be displayed.

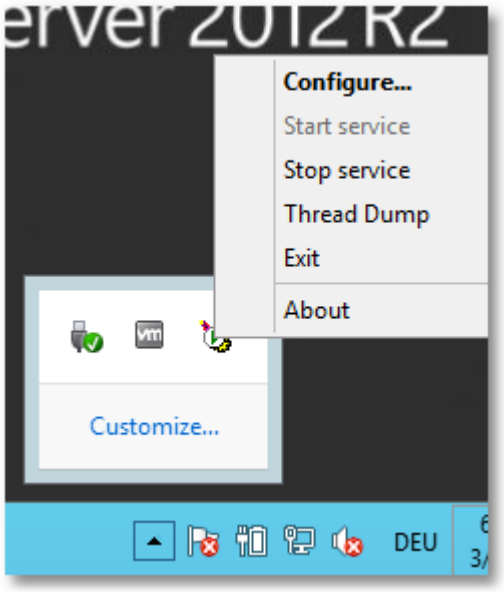
Example



2.4.3 Start and Stop Apache Tomcat

Apache Tomcat can now be stopped and started using the provided Tomcat tools and/or Windows Service panel. Available options may depend on the used Tomcat version installation.

Example

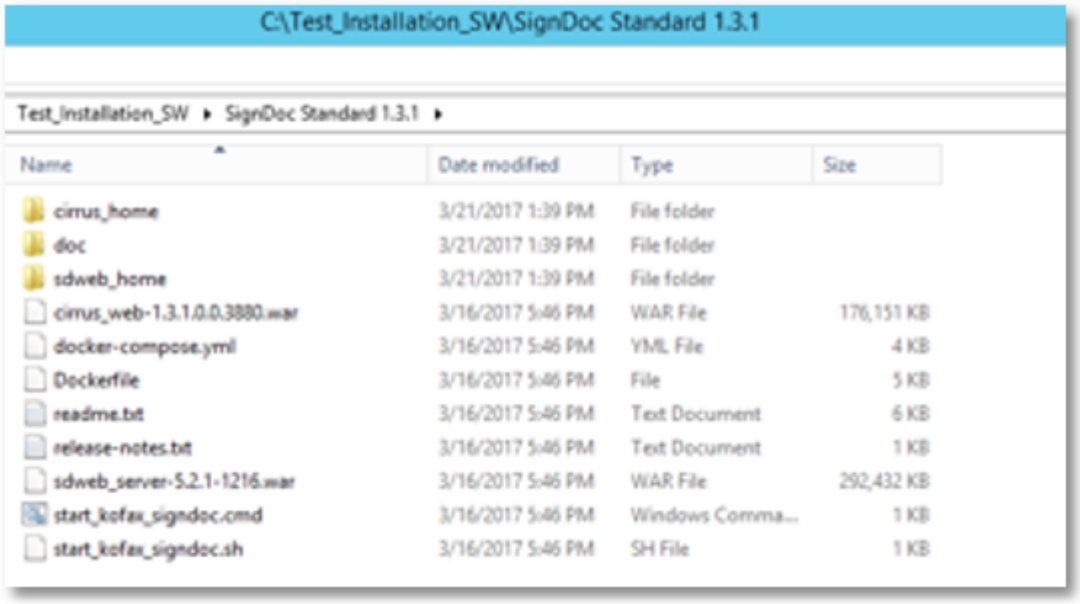


2.5 Install SignDoc Standard

SignDoc Standard consists of 2 modules that are called **SignDoc Web** and **Cirrus**.

SignDoc Web takes care of the signature process, whereas **Cirrus** takes care of user management and the Signing Package workflows. Both modules must be installed and configured.

Example shipment



sdweb_home and **cirrus_home** are configuration directories that also must be installed.

2.5.1 Create SDWEB_HOME Directory

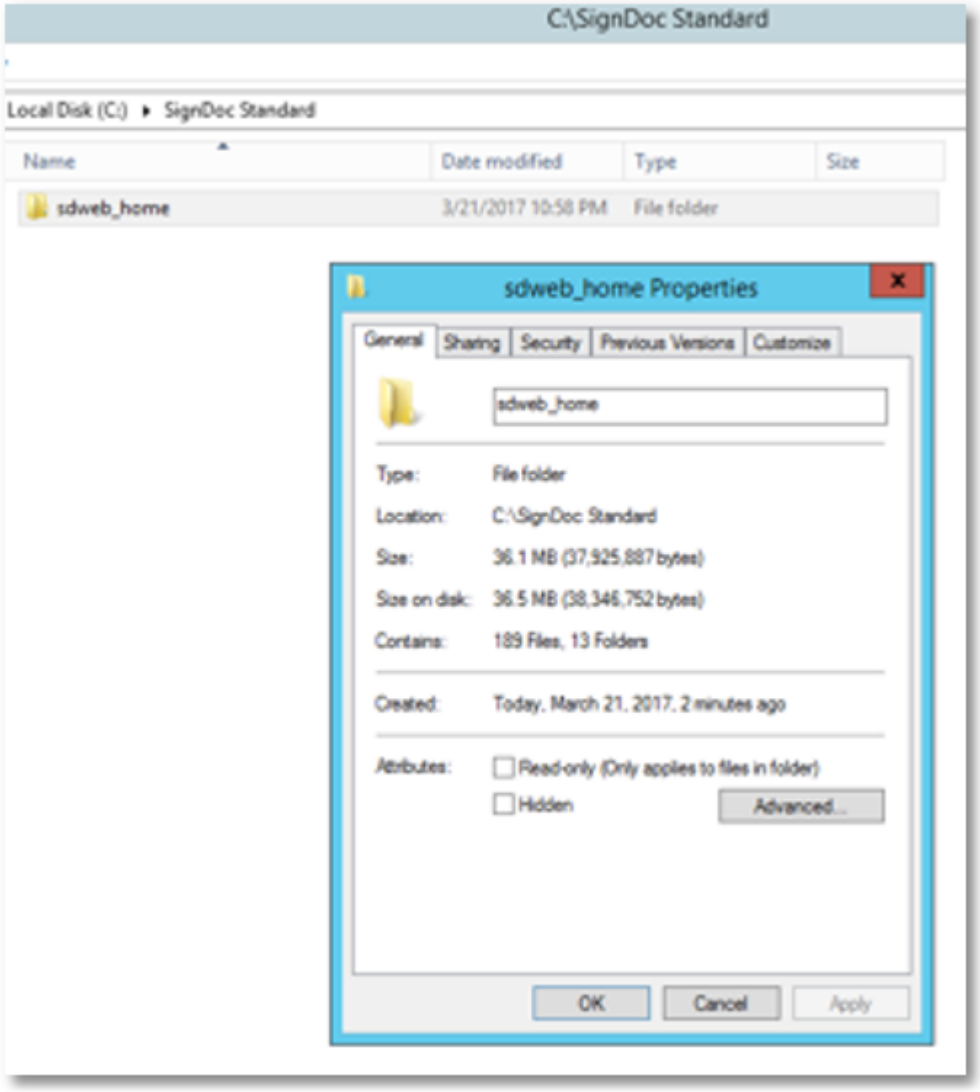
Copy the sdweb_home directory of the SignDoc Standard folder to a suitable location. Make sure that the Tomcat Service has write access to this directory and all its sub directories.

This location will be referenced as:

%SDWEB_HOME%

Example

%SDWEB_HOME%="C:\SignDoc Standard\sdweb_home"



2.5.2 Create CIRRUS_HOME Directory

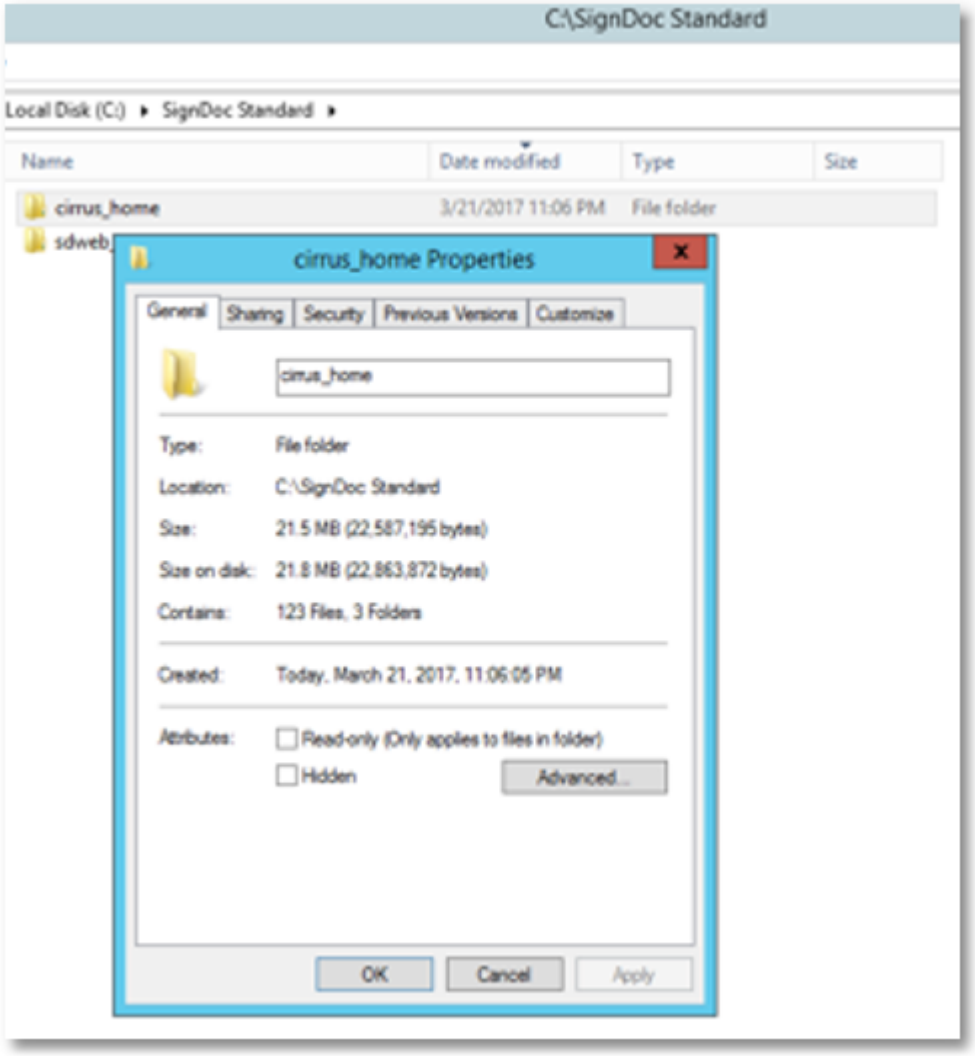
Copy the cirrus_home directory of the SignDoc Standard folder to a suitable location. Make sure that the Tomcat Service has write access to this directory and all its sub directories.

This location will be referenced as:

%CIRRUS_HOME%

Example

%CIRRUS_HOME%="C:\SignDoc Standard\cirrus_home"



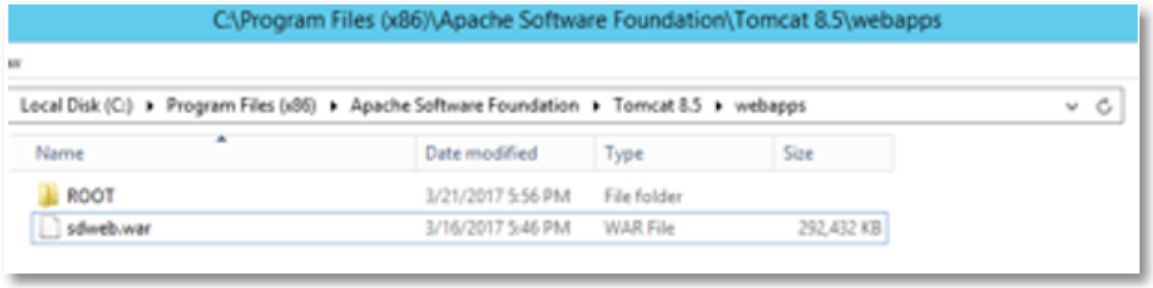
2.5.3 Install SignDoc Web Application

Copy the sdweb_server-*.war (for example sdweb_server-5.2.1-1216.war) file to %CATALINA_HOME%\webapps. Rename the file to sdweb.war.

Example

%CATALINA_HOME%="C:\Program Files (x86)\Apache Software Foundation\Tomcat 8.5"

Define %CATALINA_HOME% in your scripts or set it as system variable.



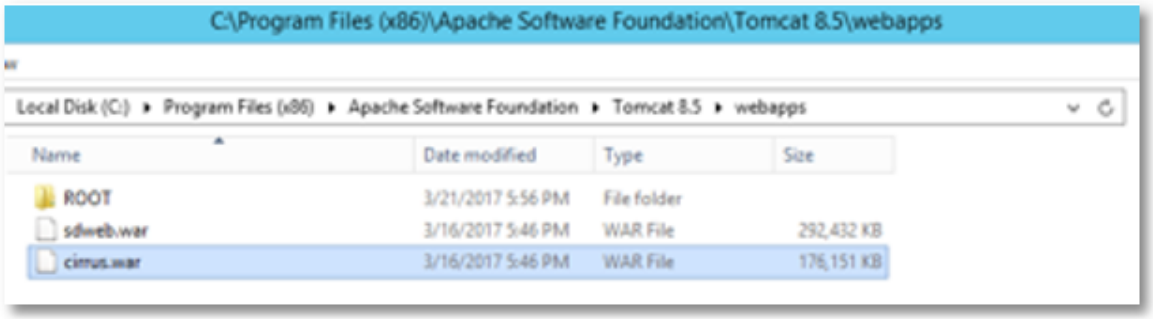
2.5.4 Install Cirrus Application

Copy the cirrus_web-*.war (for example cirrus_web-1.3.1.0.0.3880.war) file to % CATALINA_HOME%\webapps. Rename the file to cirrus.war.

Example

%CATALINA_HOME%="C:\Program Files (x86)\Apache Software Foundation\Tomcat 8.5"

If not yet done, define %CATALINA_HOME% in your scripts or set it as system variable.

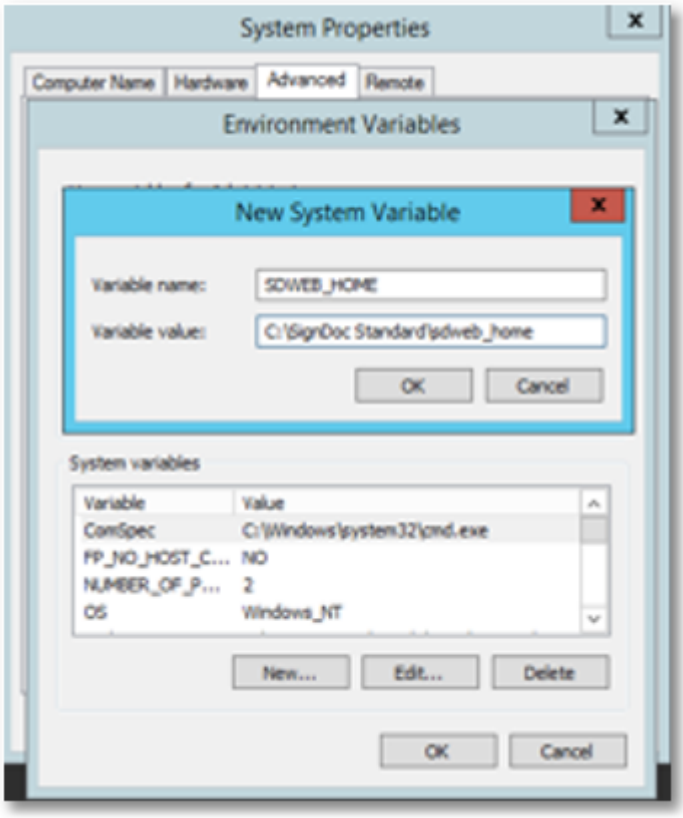


2.5.5 Set System Environment Variables

SDWEB_HOME=%SDWEB_HOME%

Replace %SDWEB_HOME% with the real path. See also chapter: Create SDWEB_HOME Directory

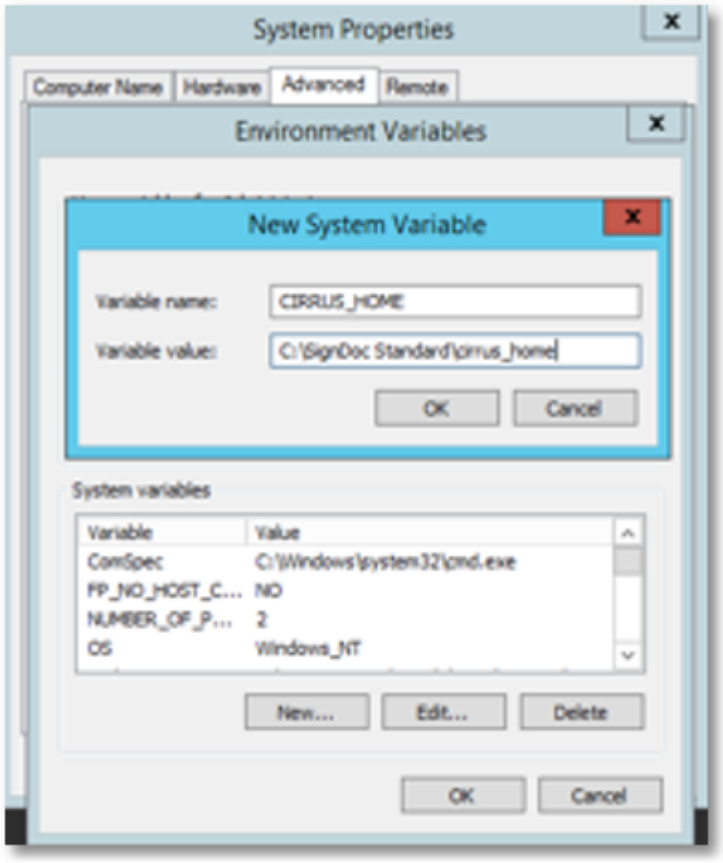
Example



CIRRUS_HOME=%CIRRUS_HOME%

Replace %CIRRUS_HOME% with the real path. See also chapter: Create CIRRUS_HOME Directory

Example

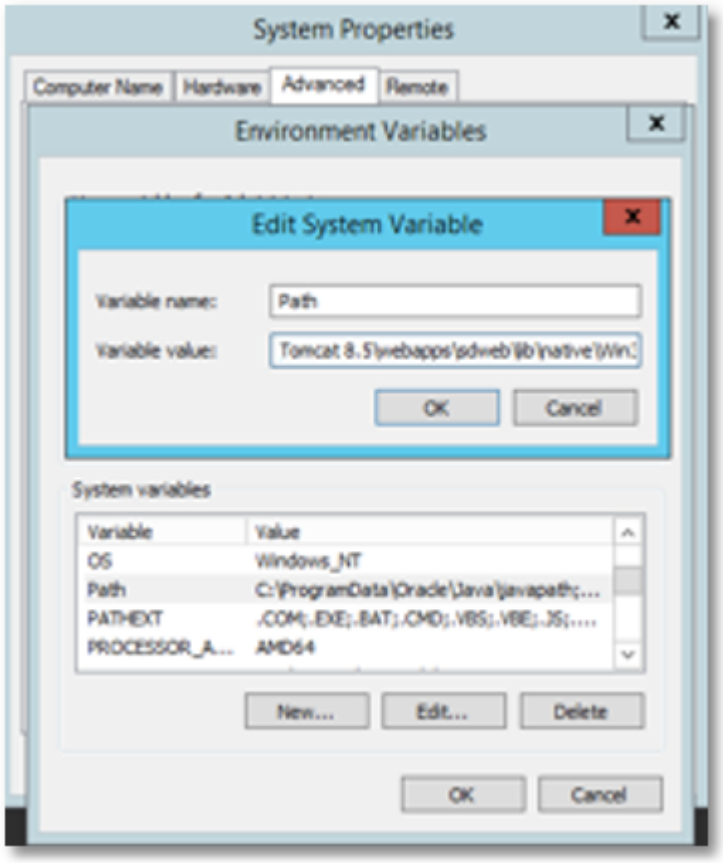


Extend the PATH Environment variable:

PATH=%PATH%;**%CATALINA_HOME%\webapps\sdweb\WEB-INF\lib\native\Win64**

Replace %CATALINA_HOME% with the real path of the Tomcat installation.

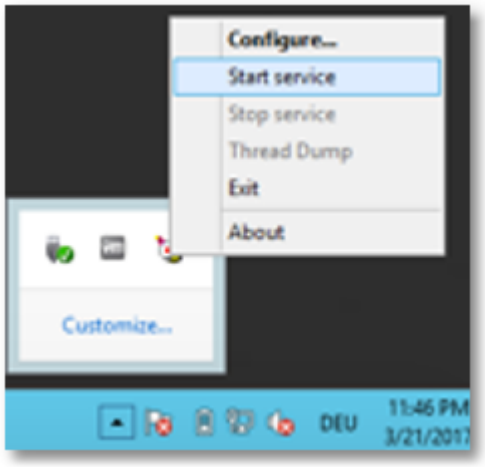
Example



2.5.6 Start the Tomcat Service

On 1st start of the Tomcat Service wait at least 1 minute.

Example



2.5.7 Open SignDoc Standard Portal Page

Open:

http://localhost:8080/cirrus

The **SignDoc Standard Portal** page should be visible.

2.5.8 Open SignDoc Web Page

Open:

http://localhost:8080/sdweb

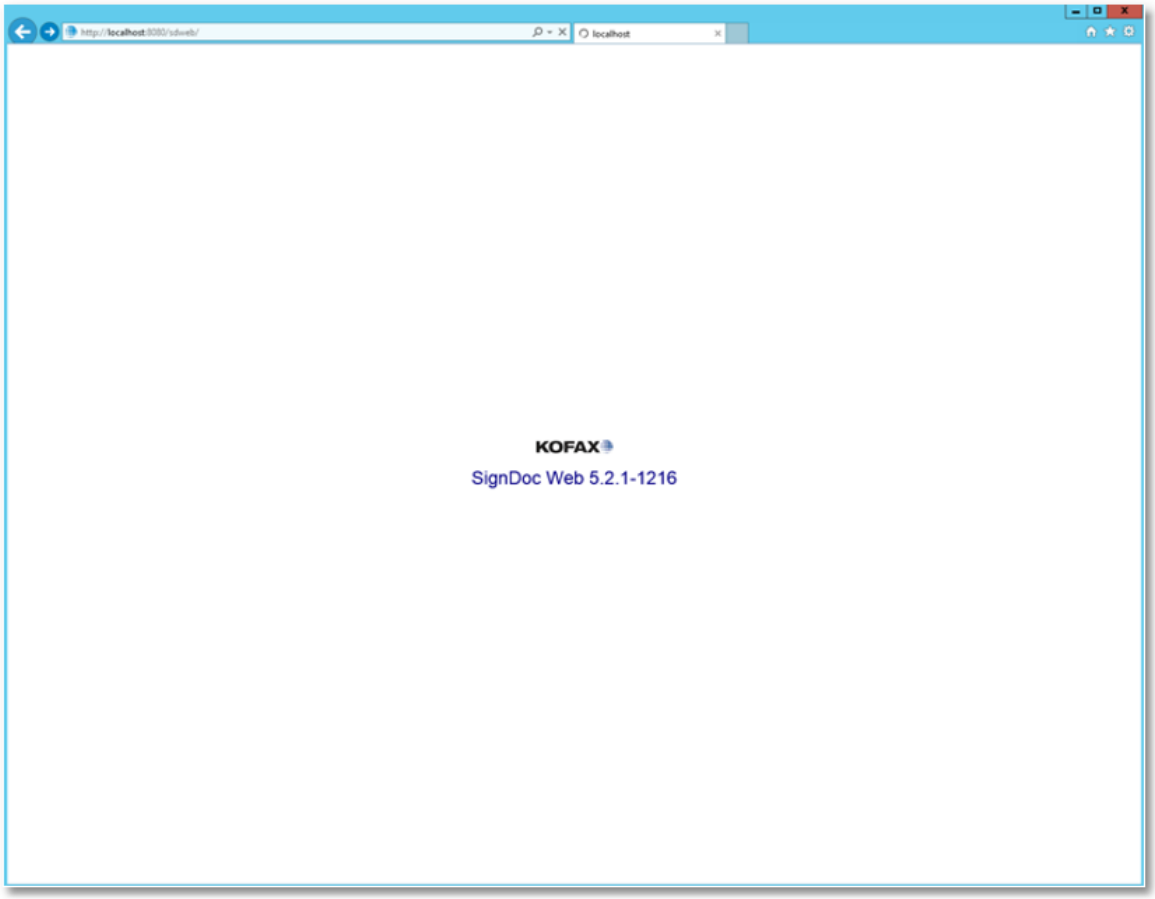
A page with a broken image link should be visible.

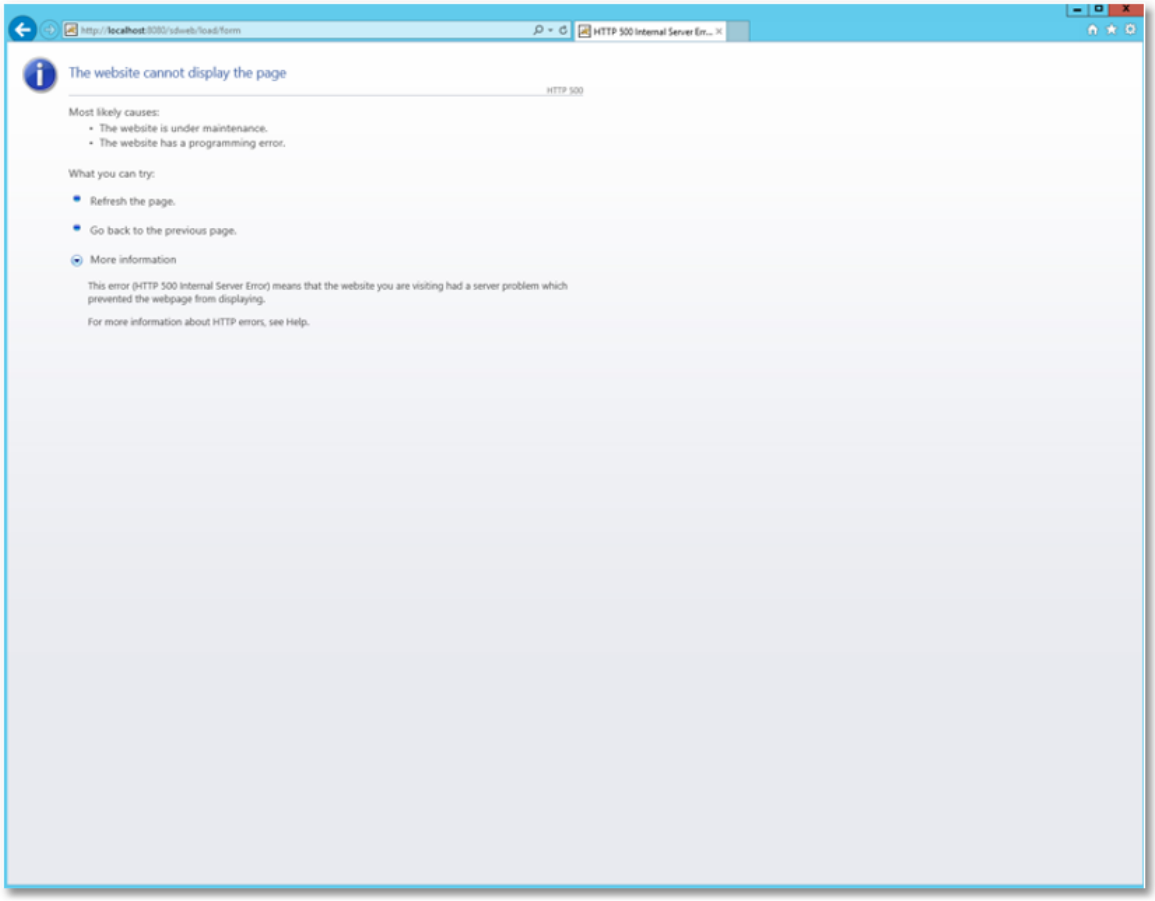
Example

NOTE

In case no broken image is displayed please verify your %PATH% settings. If the %PATH% settings are incorrect the company logo is shown shortly followed by a HTTP 500 error.

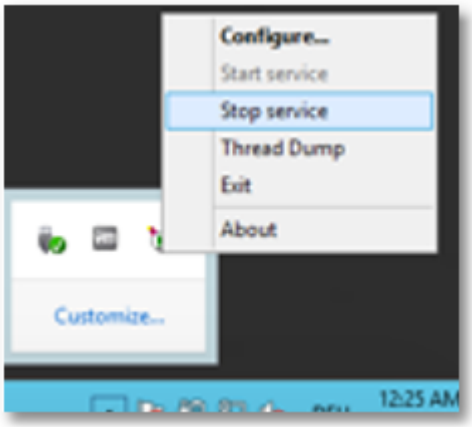
Example





2.5.9 Stop the Tomcat Service

Example



2.6 Configure SignDoc Standard

The following explains the manual steps to configure SignDoc Standard.

Additionally we provide a configuration tool. See Additional Tools.

2.6.1 Configure URLs

IMPORTANT NOTE

The file sdweb_config.groovy **must** be encoded in UTF-8.

When sdweb_config.groovy is encoded in UTF-8 BOM will result in an Internal Server Error (00001) shown in sdweb.log.

If you test the encoding in browser using calling <server>:<port>/sdweb/about page following will be displayed "It is not allowed to access the requested page!"

There are 2 types of URLs.

SPEC_EXTERNAL_SERVER_URL

This is the server's URL how users access the system. This is especially important, if a reverse proxy or load balancer is used.

Example

http://mykofaxsigndoc.com

NOTE

In this guide we do not use a reverse proxy thus SPEC_EXTERNAL_SERVER_URL is simply the DNS name of the computer plus port.

Example

http://mycomputer.mycompany.com:8080

It can also be http://localhost:8080, if the system is accessed only on the local computer)

Example

In our sample installation on one machine we replace SPEC_EXTERNAL_SERVER_URL with http://localhost:8080 in cirrus.properties and sdweb_config.groovy.

SPEC_INTERNAL_SERVER_URL

This is the server's URL how the application modules access the other modules. This is usually localhost plus port.

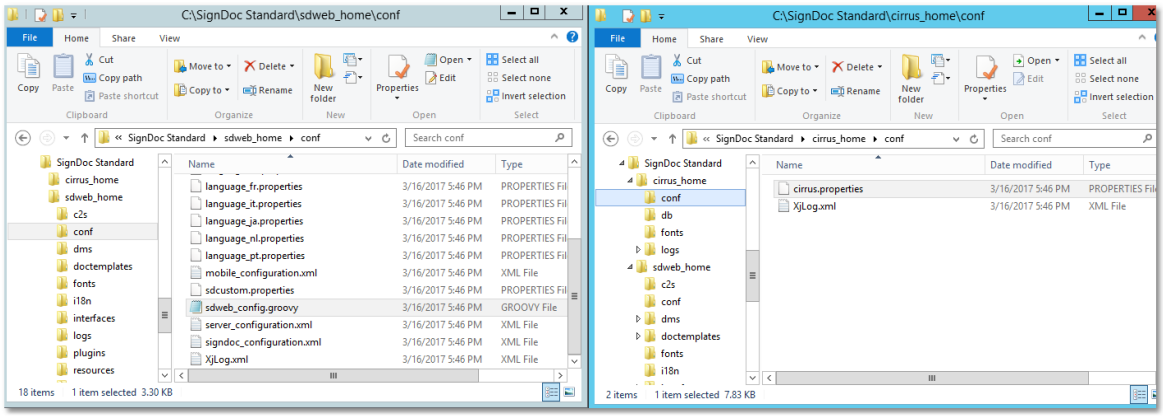
Example

http://localhost:8080

Example

In our sample installation on one machine we replace SPEC_INTERNAL_SERVER_URL with http://localhost:8080 in cirrus.properties and sdweb_config.groovy.

Open %SDWEB_HOME%\conf\sdweb_config.groovy and %CIRRUS_HOME%\conf\cirrus.properties in a text edit and replace all occurrences of SPEC_EXTERNAL_SERVER_URL and SPEC_INTERNAL_SERVER_URL with the correct values.



2.6.2 Configure Database Connection

1. Open %CIRRUS\HOME%\conf\cirrus.properties in a text editor.
2. Navigate to the DATABASE SETTINGS section and uncomment the configuration lines just below the MS-SQL Server Example.
3. There are multiple placeholders that must be replaces with real data.

   SPEC_MSSQL_SERVER -> DNS name of the MS-SQL Server

   SPEC_MSSQL_PORT -> TCP port number

   SPEC_MSSQL_DATABASE ->The database name

   SPEC_JDBC_USERNAME -> User name to access the database

   SPEC_JDBC_PASSWORD -> Password to access the database

Example

Using our example values of this guide, this would result in:

```
cirrus.spring.profiles.active=prod,ds_mssql,tx_jpa,tx_jms,ds_basic,jp
   a_openjpa,db_populate,acl_populate,jms_basic,jms_activemq,acl_perm
   ission,acl_authorization,swagger
jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver
# replace <sql-node> with localhost when following the sample
jdbc.url=jdbc:jtds:sqlserver://<sql-node>:1433/signdoc
jdbc.username=signdoc
jdbc.password=2beChanged!
```


2.6.3 Configure Email Settings

SignDoc Standard requires a reliable SMTP server to be able to send out notifications, invites and Signing Packages.

Open %CIRRUS\HOME%\conf\cirrus.properties in a text editor.

Navigate to the EMAIL SETTINGS section and configure the SMTP properties that are required to access the mail server. There are examples for mailcatcher, a simple email relay and an TLS based configuration that is usable with AWS-SES.

Sample using mailhog as mail application:

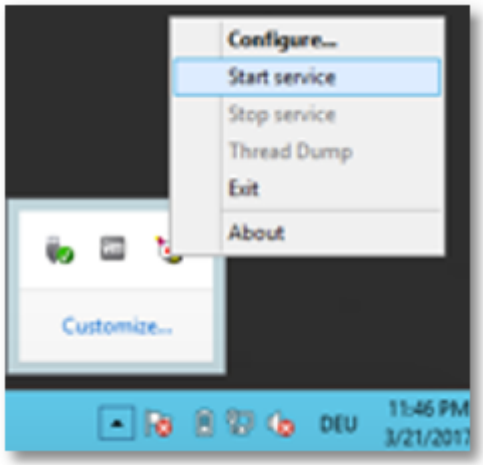
```
############################
# A local SMTP service (mailhog)
# To use start a mailcatcher service
# MailHog: https://github.com/mailhog
# MailHog requires no installation an can be downloaded here:
   https://github.com/mailhog/MailHog/releases
mail.smtp.host=localhost
mail.smtp.port=8025
mail.smtp.user=dont_reply@mailhog
mail.smtp.from=dont_reply@mailhog
# Do not set a password if the server supports no authentication
#mail.smtp.password=
############################
```

[optional] Send Startup email

SignDoc Standard can send an email to a predefined email address every time it starts. Look for cirrus.startup.email and set a valid email address that will be notified whenever the server is started.

2.6.4 Start the Tomcat Service

Example



2.6.5 Open the Administration Center

Open the URL of the **Administration Center** to create accounts and users:

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

For this guide this is:

http://***<app-node>:8080***/cirrus/admin-center

Example

When

<app-node> is localhost

the URL is

http://localhost:***8080***/cirrus/admin-center

Please continue with the *Kofax SignDoc - Server Administrator's User Guide* to create accounts and users.

2.7 Additional Tools (optional)

tools/setup_config_files.cmd

This command file sets the most important information of the different configuration files. It will use the files %CIRRUS_HOME%\template_cirrus.properties and %SDWEB_HOME%\template_sdweb_config.groovy as template for the new config files and will replace specific keywords with predefined values. Note that this tool does not handle the mail configuration in %CIRRUS_HOME%\cirrus.properties.

Specifically, it sets the values of these placeholder keywords:

SPEC_EXTERNAL_SERVER_URL

SPEC_INTERNAL_SERVER_URL

SPEC_MSSQL_SERVER

SPEC_MSSQL_PORT

SPEC_MSSQL_DATABASE

SPEC_JDBC_USERNAME

SPEC_JDBC_PASSWORD

SPEC_STARTUP_EMAIL

Usage

- Open the Setup_config_files.cmd with a text editor. Optional: Have a look at the script code.
- Look at the section between “rem configuration property values - start” and “rem configuration property values - end”.
- Adjust the values in this section as they are needed for your environment.
- Save the file.
- Before executing the file, be sure that the SDWEB_HOME and CIRRUS_HOME environment variables are set correctly.
- Execute the file. Before the configuration files are newly written, a backup copy (filename ends with _backup) will be created.

3 Advanced Installation

3.1 Integrate with Kofax TotalAgility

To successfully connect Kofax SignDoc to Kofax TotalAgility the below parameters need to be set in the **cirrus.properties** file.

Resource	Description
`cirrusplugins.kta.notifications.enabled`	Enable Kofax SignDoc callbacks to Kofax TotalAgility. Possible values: true, false Must be set to "true" in order for sending notifications to Kofax TotalAgility.

Resource	Description
`cirrusplugins.kta.session.id`	The Kofax TotalAgility Session Id, authenticates Kofax SignDoc against Kofax TotalAgility
`cirrusplugins.kta.url`	The Kofax TotalAgility URL
`winword.field.identification.key`	The (unique) oid of the newly created signer can be set either from the "Suggested signer" value of the MS Word Signature Line or from an internal signatureline id (which is also used for the created field name). If you specifiy field_name then the signatureline id is used for the field name and signer oid. If you specify signer_name (default value) then the entered value of "Suggested signer" is used for the signer oid. **IMPORTANT NOTE** The Kofax TotalAgility integration requires the value to be field_name.

IMPORTANT NOTE

1. When using a Kofax SignDoc package template together with Kofax TotalAgility it needs to have a signer included.
2. When using a PDF document which already has signature fields they all have to have an associated signer.
   If they remain assigned to "Any", they will not be visible to Kofax TotalAgility.

Kofax TotalAgility Account Specific Configuration

With Kofax SignDoc 1.3 the Kofax TotalAgility link configuration can be set in two ways:

- The same way as in Kofax SignDoc 1.2, by setting the parameters 'cirrusplugins.kta.notifications.enabled', 'cirrusplugins.kta.session.id' and 'cirrusplugins.kta.url' in **cirrus.properties**, or,
- By using the REST API configuration interface and set the configuration on an account basis.

With Kofax SignDoc 1.2 the whole installation had to be linked with Kofax TotalAgility, or not.

With the second configuration method it is possible to link only individual accounts with the Kofax TotalAgility system, or link individual accounts with different Kofax TotalAgility systems. This is particularly useful in a cloud deployment where different accounts can use different systems, or not be linked to Kofax TotalAgility at all.

To link one account with a Kofax TotalAgility system you have to set following settings via the REST API configuration interface:

Setting	Description
`cirrus.kta.notifications.enabled`	Enable / disable the linking with a Kofax TotalAgility system (true / false)

Setting	Description
`cirrus.kta.url`	The URL where the Kofax TotalAgility system can be reached
`cirrus.kta.sessionid`	The Kofax TotalAgility session id to link with.

Please refer to the REST API configuration interface documentation in *Kofax SignDoc - Developer's Guide* on how to set the settings. The settings can be set globally for the whole system, if you do not provide an account id, or for a specific account id that you provide as part of your REST request.

Each account can link to a different Kofax TotalAgility system, or to none. The settings take effect immediately and do not need a system reboot.

3.2 Authentication LDAP

General

This section describes the specification of the basic authentication via LDAP provided with Kofax SignDoc 1.3.

Prerquisites

The LDAP support in Kofax SignDoc is not usable and not supported in a multi-tenant environment. Kofax SignDoc maps LDAP user entries to Kofax SignDoc users by the unique email address. A Kofax SignDoc 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.

Information: 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**.

NOTE

Activate LDAP only *after* you have created the single account.

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 Kofax SignDoc. Kofax SignDoc maps LDAP attributes to Kofax SignDoc user attributes. Each name of an LDAP attribute has a default value but can be customized by a Kofax SignDoc property:

Kofax SignDoc property	Default value	Mapped to this Kofax SignDoc user attribute	Constraints
ldap.user.name.attr	cn	User name	
ldap.user.mail.attr	mail	email	(mandatory setting) Must not already exist as a Kofax SignDoc user. The email address must match this regular expression: ^[A-z0-9\\._%+\\-]+@[A-z0-9\\.\\-_]+
ldap.user.uid.attr	uid	OID	(optional setting) Must not already exist and match the Kofax SignDoc validation rule for OID (regex ^[a-zA-Z0-9_\-]+$)

NOTE

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

Configuration File cirrus.properties

As usual the default values of LDAP related properties are located in the classpath resource

```
CIRRUS_HOME/conf/cirrus.properties
```

All values can be customized by the **cirrus.properties** file at CIRRUS_HOME directory.

Property	Type	Description
authentication.provider	String	Activates LDAP support. Must be 'LDAP,CIRRUS'
ldap.url	String	The URL to connect to an LDAP server. **NOTE** It is strongly recommended to use the ldaps protocol because passwords are send in plain text over the network. A suitable certificate must be installed at the server in that case. **Example** `ldap://ad.kofax.com:389/dc=kofax,dc=de`
ldap.manager.dn	String	The manager DN. If your LDAP implementation does not allow anonymous access a suitable user and password must be defined here. **Example** `uid=admin,ou=system`

Property	Type	Description
ldap.manager.password	String	The manager password. **Example** `ldap.manager.password=secret` Omit manager dn and password for anonymous access.
ldap.userdn.patterns	String	The value is a list of distinguished names (DN) separated by a colon. **NOTE** Because the field delimiter is the colon (':'). A DN containing colon(s) must be double-quoted. And a double-quoted DN must escape any double-quote sign with the escape character '\', should it be present in the DN. **Example** `uid={0},ou=Users` The key '{0}' will be substituted with the login name.
ldap.user.search.base	String	The base DN for starting a search. **Example** `dc=kofax,dc=de`
ldap.user.search.filter	String	A filter for the search (see RFC 2254) **Example** `(cn=Babs Jensen)`
ldap.user.name.attr	String	The LDAP attribute which maps to a Kofax SignDoc user name. **Example** `ldap.user.name.attr=cn` Default: cn
ldap.user.mail.attr	String	The LDAP attribute which maps to a Kofax SignDoc user email. **Example** `ldap.user.mail.attr=mail` Default: mail

NOTE

Additional “Brute Force Authentication Prevention” is not implemented if LDAP Authentication is configured.

3.3 CSRF Protection via Header Matching

Kofax SignDoc 1.3 includes CSRF (cross-site request forgery) attack protection by comparing the referer headers to the machine name Kofax SignDoc is deployed on. For this mechanism to work, the setting 'cirrus.external.url' in **cirrus.properties** file has to be configured with the correct URL your installation will be reachable by. Only requests via this URL will be allowed. Alternative machine names can be rejected, as well as links from other pages that do not match one of the standard entry points.

In case you want do disable CSRF protection you can set:

```
cirrus.csrf.headers.enabled=false
```

in **cirrus.properties** file.

4 Upgrading Kofax SignDoc

To upgrade an existing Kofax SignDoc 1.2 system to 1.3.0, following steps need to be performed:

- Stop the Kofax SignDoc 1.2 system
- Backup the database (strongly advised)
- Deploy the Kofax SignDoc 1.3 files
- Migrate the database using the built in migration (depending on the configuration, this step can be automatic)
- Update / change the configuration (if necessary)
- Start the Kofax SignDoc 1.3 system
- Add additional configuration via the REST configuration interface if needed

To upgrade an existing Kofax SignDoc 1.3.0 system to 1.3.1, following steps need to be performed:

- Stop the Kofax SignDoc 1.3.0 system
- Back up the database
- Deploy the Kofax SignDoc artifacts
- Migrate the database
- Update / change the configuration
- Start the Kofax SignDoc system
- Add additional configuration

4.1 Upgrade Steps Kofax SignDoc 1.2 to 1.3.0

Stop the Kofax SignDoc 1.2 system

The upgrade can't be performed while the system is running. Kofax SignDoc 1.2 servers have to be shut down for the upgrade.

Backup the database

Even though this step is not strictly necessary we strongly advise you to backup your database. This way you can revert to the previous state in case something goes wrong.

Deploy the Kofax SignDoc artifacts

Depending on your installation you would need to extract your files for a local installation, or deploy a Kofax SignDoc Docker container to the location / system of your choice. Please refer to the installation documentation that matches your deployment scheme.

Migrate the database

Your system can be configured to either perform an automatic database migration or use a manual procedure. The later is recommended in case of a system that is running more than one server.

In case your system is configured to perform an automatic migration, the database migration will be performed automatically during the first start of the application.

In case your system is configured to perform a manual migration you would need to trigger the database migration now. For a standalone migration you would have to perform the 'flyway migrate' command, a docker container configured with Flyway can perform the migration by calling the 'flyway' entry point with 'flyway migrate'. If Flyway is not included in the deliverables, it can be downloaded and installed from www.flywaydb.org.

Please consult Database Migration section, depending on the migration deployment you configured with Kofax SignDoc 1.2.

Update / change the configuration

The default system configuration changed from Kofax SignDoc 1.2 to 1.3.0. In case you deploy the default configuration files that come with the new system those configuration files should already contain the necessary changes.

If you deploy customized configuration files, you should be aware of the following changes:

- SignDocWeb to Kofax SignDoc plugin configuration:
  The signing part of the application interacts with Kofax SignDoc via a set of plugins defined in **sdweb.config.groovy**. If you use a customized **sdweb.config.groovy**, please ensure that the section named 'Cirrus plugin configuration' is included as such and the setting 'cirrus.rest.url' points to the V4 endpoint of the cirrus (KSD) installation.
- CSRF protection:
  The application tries to protect against CSRF attacks by checking request headers. The referer headers need to match the machine name configured for your server. They are checked against the value configured in the 'cirrus.external.url' setting in cirrus.properties. In case this setting is not configured, the CSRF header validation is disabled.
- Kofax SignDoc system linking
  In addition to the Kofax SignDoc 1.2 configuration regarding the Kofax Total Agility interface, which allowed you to configure the Kofax TotalAgility link on a global basis, it is now possible to only link individual accounts with Kofax TotalAgility. If this is desired, the Kofax TotalAgility settings have to be removed from **cirrus.properties** and the configuration has to be performed after system start via the REST configuration interface. A description is available in the Kofax TotalAgility Account Specific Configuration section of the manual.

Start the Kofax SignDoc system

Start the Kofax SignDoc system as you would do for a normal installation and verify it is running correctly.

Add additional configuration

Some new configuration settings can be configured via the REST configuration interface. These settings can be set globally, or on an account basis. This enables you to set some settings for a specific account, or enable / disable a feature on an account basis. The change requires a running system. The settings are effective immediately and do not need a system restart. Please refer to the *Kofax SignDoc - Developer's Guide*, section "Configuration Requests" for a detailed setting description.

- SMS notification
  In case you want to use the SMS notification feature to enable two factor signer authentication, you can enable the appropriate plugin via the REST interface. Please refer to the *Kofax SignDoc - Developer's Guide*, section "Configuration Requests" on how to do so. The SMS server can be configured globally or different settings can be used for each account.
- Kofax SignDoc system linking
  The interface to a Kofax TotalAgility system can be configured on an account specific basis. Thus only individual accounts can be linked to a Kofax TotalAgility system (as opposed to the whole installation in Kofax SignDoc 1.2).

4.2 Upgrade Steps Kofax SignDoc 1.3.0 to 1.3.1

Stop the Kofax SignDoc 1.3.0 system

The upgrade can't be performed while the system is running. Kofax SignDoc 1.3.0 servers have to be shut down for the upgrade.

Back up the database

Even though this step is not strictly necessary we strongly advise you to back up your database. This way you can revert to the previous state in case something goes wrong.

Deploy the Kofax SignDoc artifacts

Depending on your installation you would need to extract your files for a local installation, or deploy a Kofax SignDoc Docker container to the location / system of your choice. Please refer to the installation documentation that matches your deployment scheme.

Migrate the database

Your system can be configured to either perform an automatic database migration or use a manual procedure. The latter is recommended in case of a system that is running more than one server. In case your system is configured to perform an automatic migration, the database migration will be performed automatically during the first start of the application.

In case your system is configured to perform a manual migration you would need to trigger the database migration now. For a standalone migration you would have to perform the 'flyway migrate' command, a docker container configured with Flyway can perform the migration by calling the 'flyway' entry point with 'flyway migrate'. If Flyway is not included in the deliverables, it can be downloaded and installed from www.flywaydb.org.

Please consult Database Migration section, depending on the migration deployment you configured with Kofax SignDoc 1.3.0.

Update / change the configuration

The default system configuration changed from Kofax SignDoc 1.3.0 to 1.3.1. In case you deploy the default configuration files that come with the new system those configuration files should already contain the necessary changes.

If you deploy customized configuration files, you should be aware of the following changes:

- SignDocWeb to Kofax SignDoc plugin configuration.
  In case the new feature to digitally sign documents by a trusted service provider will be used, the plugin required to contact the TSP has to be added to the list. This is defined in **sdweb_config.groovy**.
- REST API versions used.
  - The current REST API version for SignDoc Web is now v4.
    Please check the setting **sdweb.rest.url** in **cirrus.properties**.
  - The current REST API version for Kofax SignDoc (Cirrus) is now v5.
    Please check **cirrus.rest.url**, or the individual plugin urls in **sdweb_config.groovy**.

Start the Kofax SignDoc system

Start the Kofax SignDoc system as you would do for a normal installation and verify it is running correctly.

Add additional configuration

Kofax SignDoc 1.3.1 adds the ability to have documents digitally signed by a trusted service provider. This feature has been implemented as a plugin, letting the user chose which trusted service provider is used on an account basis.

The plugin configuration is done via the configuration API REST interface. Please refer to the TSP plugin configuration section in the *Kofax SignDoc - System Administration* guide for the options available if you want to use this feature.

5 Database Migration

This chapter describes the database migration mechanism used by SignDoc starting with version 1.1.0.1.

5.1 Overview

Any product that uses a schema based database and gets past its first version faces the problem of tackling database changes while migrating from one version to another. This includes changes to the database schema, like adding a new column, moving data from a location to another etc. We have not only to adapt the database to the new schema, we also have to migrate the existing data to fit it.

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

5.1.1 Flyway

SignDoc uses Flyway to standardize database migration scripts. You can read about flyway at http://flywaydb.org/. In short (please check the Flyway documentation at the 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, one can always tell what state the database is in and what changes still need to be applied.

Flyway migrations can either be run from the command line, or be integrated into the product itself. 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.

5.2 Flyway Use in SignDoc

5.2.1 Integration and Configuration

Flyway is built into SignDoc, starting with release 1.1.0.1. Each time SignDoc is started, it will check if the database is up to date and can run any outstanding database migrations.

Migrations can be configured to run either automatically or manually. Running migrations automatically can be convenient for a classic deployment with a single server, or a test environment with frequent changes. In a cloud environment running multiple servers a manual invocation of the database migration is recommended. This allows for a better control of the process, including the necessary backup and QA steps.

The way migrations are run is controlled by the `cirrus.migrations.enabled property`:

`true`	Enables automatic migrations. The application will compare the version currently stored in the database and attempt to migrate it to the one used by the application. It will apply all necessary steps in sequence, without needing confirmation.
`false`	Disables automatic migrations. The application will compare the version currently stored in the database and refuse to start if it does not match the one used by the application (will throw an exception). The migration step has to be run manually using the command line tool or Docker container.

Regardless of the setting used it is **strongly recommended to 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.

5.2.2 Version Numbers

Following schema is used for SignDoc migration version numbers:

<ver major>.<ver minor>.<release>.<bugfix>.<hotfix>_<migration sequence>

The migration also includes a short textual description. A sample output of the version information is shown below:

```
+--------------+---------------------------+---------------------+---------+
| Version      | Description               | Installed on        | State   |
+--------------+---------------------------+---------------------+---------+
| 1.1.0.0.0.0  | Baseline                  | 2015-11-24 17:56:54 | Success |
| 1.1.0.1.0.1  | Upgrade                   | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.1  | Release upgrade           | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.2  | NewAccountLicenseHandling | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.3  | Add package counter       | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.4  | AddDnsLabel               | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.5  | AddTimeZoneToAccount      | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.6  | RemoveUserStateINACTIVE   | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.7  | AddContactInfoToAccount   | 2015-11-24 17:56:54 | Success |
| 1.2.0.0.0.8  | HandleKeysTable           | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.9  | AddSignatureSettings      | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.10 | AddAccountPersonalization | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.11 | UserRolesNotNullable      | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.12 | DropObsoleteTimestamps    | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.13 | Add document counter      | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.14 | RemoveUnusedUserStates    | 2015-11-24 17:56:55 | Success |
| 1.2.0.0.0.15 | NotDeleteAuditTrails      | 2015-11-24 17:56:55 | Success |
+--------------+---------------------------+---------------------+---------+
```

5.2.3 Classic Deployment

The classic deployment describes the installation of SignDoc in a servlet container (Tomcat), without the use of Docker containers.

5.2.3.1 Automatic Migration

To enable automatic migrations you have to set `cirrus.migrations.enabled` to `true` in **cirrus.properties**. SignDoc will check the database version and run pending migrations automatically during system start.

5.2.3.2 Manual Migration

If is set to `false`, SignDoc will only verify if the database has been updated to the current version. Migrations have to be run manualy using the flyway command line tool.

5.2.3.2.1 Command Line Tool

The *flyway* command line tool can be used to query the database version information, check outstanding migrations, perform migrations and clean or repair the database. All necessary information, like database driver, URL, login info, etc. can be given as arguments. A more convenient way is to store them in a configuration file. A sample configuration (**flyway.conf**) is shown below:

```
# Database URL
flyway.url=jdbc:jtds:sqlserver://servername/database_name
```

```
# User to use to connect to the database (default: <<null>>)
flyway.user=username
# Password to use to connect to the database (default: <<null>>)
flyway.password=password

# Locations starting with filesystem: point to a directory on the
   filesystem and may only contain sql migrations.
flyway.locations=classpath:sql/migration/net_sourceforge_jtds_jdbc_Dr
   iver,classpath:sql/migration/common,classpath:de/softpro/cirrus/db
   /migrations

# Comma-separated list of directories containing JDBC drivers and
   Java-based migrations. (default: <INSTALL-DIR>/jars)
# flyway.jarDirs=<path to flyway>/flyway-3.2.1/jars,<path to cirrus-
   db-<verionnumber> directory>
flyway.jarDirs=/flyway/jars,/tomcat/webapps/cirrus/WEB-INF/lib

# The version to tag an existing schema with when executing baseline.
   (default: 1)
flyway.baselineVersion=1.1.0.0.0_0
# Whether to automatically call baseline when migrate is executed
   against a non-empty schema with no metadata table.
flyway.baselineOnMigrate=true
```

Fields that have to be configured are highlighted yellow.

You can check the status of the database using the command:

```
flyway info
```

Migrations can be applied using:

```
flyway migrate
```

The *flyway* command line and the flyway web page provide a reference on the available commands and options.

If Flyway is not included in the deliverables, it can be downloaded and installed from www.flywaydb.org.

5.2.4 Docker Deployment

In case of a Docker deployment, the complete application environment is packaged inside a Docker container. This includes the migration tool. The default migration setting for a Docker container is `false`.

5.2.4.1 Automatic Migration

If automatic migrations are desired the migration property can be set to true by providing the environment variable

```
SPEC_CIRRUS_MIGRATIONS=true
```

at container start:

```
docker run –e SPEC_CIRRUS_MIGRATIONS=true … softpro/signdoc-
   standard:<version>
```


5.2.4.2 Manual Migration

Running migrations manually is the default setting for the Docker container. Migrations can be run by overriding the **flyway.conf** and invoking flyway migrations. It is generally recommended to add a Docker layer to the container that copies the **flyway.conf** file into a configured container version (to set the database URL and credentials). The sample shown above lists the configuration needed for the Docker container (notably the **flyway.jardirs** setting). A configured container can be run using:

```
docker run -ti softpro/signdoc-standard:<version> flyway <command>
```

6 FAQ

General

- **Useful logfiles for support requests**
  - CIRRUS_HOME\logs\cirrus.log
  - CIRRUS_HOME\logs\error\{ErrorContext}.log - only if error occurred
  - SDWEB_HOME\logs\sdweb.log
  - SDWEB_HOME\logs\error\{ErrorContext}.log - only if error occurred
  - TOMCAT_FOLDER\logs*.log
  - Windows Event Log
  - Firewall Logs
  - Browser Console Log

Email Settings

- **An error occurred sending an email**
  - You specified the wrong server. The server you specified exists, but it is not an SMTP server.
  - You specified the wrong port number. Ask whoever runs the SMTP server what the correct port number is.
  - The server is down. This is usually temporary. If it persists, contact whoever administers the server.
  - Your firewall is blocking the port.
  - Your ISP is blocking the port. This usually affects port 25, and you can often work around it by using port 587, but details depend on your ISP and on the SMTP server's configuration.
  - You specified TLS, but the server does not support it.

Apache Tomcat

- **Security**
  - For a production environment restrict the communication between cirrus and sdweb to local interfaces.
- **Performance**

- The connector default size in bytes for POST requests is limited to 2 Megabytes, increase the size to handle larger documents by adding the attribute *maxPostSize="52428800"* to the tomcat connector definition.

7 Appendix

7.1 Contact Information

The Kofax technical support team will be happy to assist you.

If you need support with regards to purchased Signature products please contact us via Kofax Customer Portal:

https://techsupport.kofax.com

7.2 Trademarks

- SignDoc, SignPlus, FraudOne, SignWare, SignInfo, SignCheck, SIVAL, SignPad, SignSecure are registered trade marks of Kofax Inc. or its affiliates – all rights reserved.
- All other brand and product names can be trademarks, service marks or other intellectual property of the respective owners who reserve all related rights.

7.3 Copyright Notice

Index

A

C

D

F

H

I

K

L

S

U