



Workflow Accelerator Administrator Guide

3.146

Contents

1	Introduction.....	8
1.1	Software components.....	8
1.1.1	Workflow Accelerator system.....	8
1.1.2	Apache Tomcat.....	8
1.1.3	MongoDB.....	8
1.1.4	Node.js.....	9
1.1.5	Mail server (SMTP).....	9
1.1.6	Web browser.....	9
1.2	Naming conventions.....	9
2	Installing Signavio Workflow Accelerator.....	10
2.1	Unpack Signavio Workflow Accelerator.....	10
2.2	System requirements.....	10
2.2.1	Operating system.....	10
2.2.2	Hardware requirements.....	11
2.2.3	Other systems.....	12
2.2.4	Software requirements.....	12
2.2.5	Web browser.....	12
2.3	Installing Java.....	13
2.4	Installing Tomcat.....	13
2.4.1	Windows.....	13
2.4.2	Debian.....	13
2.5	Configuring Tomcat.....	14
2.5.1	Java options.....	14
2.5.2	Windows.....	15
2.5.3	Debian.....	15
2.5.4	Set up the Tomcat connector.....	16
2.6	Installing the Workflow Accelerator web application.....	16
2.7	Installing MongoDB.....	17
2.7.1	Remarks for downloading the binaries.....	17

2.7.2	Windows	17
2.7.3	Debian	19
2.8	Configuring MongoDB	19
2.8.1	Add a database user for Workflow Accelerator	21
2.8.2	SSL setup	25
2.8.3	Backup and restore	29
2.9	Installing Node.js	29
2.9.1	Windows	29
2.9.2	Debian	29
2.10	Installing the Workflow Accelerator script engine	30
2.10.1	Wrapping the script engine into a service	31
2.11	Installing the Workflow Accelerator mail relay	35
2.11.1	Create an email domain for the mail relay	35
2.11.2	Install and configure the mail relay	36
2.11.3	Configure the web application	39
2.11.4	Testing the mail relay	39
2.12	Configuring Workflow Accelerator	41
2.12.1	Install the license file	41
2.12.2	Update the Workflow Accelerator configuration file	41
2.12.3	Configure logging	44
3	Optional configuration	45
3.1	User and group synchronisation with Active Directory	45
3.1.1	Creating the Active Directory groups	46
3.1.2	Creating a new configuration	46
3.1.3	Validating a configuration	47
3.1.4	Synchronising users and groups	47
3.2	Single Sign-On using Windows and Kerberos/SPNEGO	47
3.2.1	Install the SPNEGO library	48
3.2.2	Configuring the SPNEGO SSO Servlet filter	48
3.2.3	Adding SPNEGO configuration files	49
3.2.4	Registering Tomcat's Service Principal Name (SPN)	50

3.2.5	Configuring the Tomcat user	51
3.2.6	Troubleshooting	51
3.3	Configure Google service actions	52
3.3.1	Create OAuth 2.0 client credentials for Google services	53
3.3.2	Activate required Google APIs	53
3.3.3	Download and install the client credentials	54
3.3.4	Apply for application verification	55
3.4	Implement a configuration provider	55
3.4.1	Create your own configuration provider	55
3.4.2	Configure Workflow Accelerator to use your configuration provider	56
4	Signavio Process Manager integration	57
4.1	Compatibility list	57
4.2	Setup	58
4.3	User management	60
5	Updating Signavio Workflow Accelerator	61
5.1	Unpack Signavio Workflow Accelerator	61
5.2	Stop Workflow Accelerator	62
5.3	Backup MongoDB	62
5.4	Backup the configuration files	62
5.5	Install a new version	63
5.6	Revisit the configuration files	64
5.6.1	Reconfigure your optional configuration provider	64
5.7	Start Workflow Accelerator	64
5.8	Restore an older version	65
6	Update notes for specific versions	66
6.1	Updating to Signavio Workflow Accelerator 3.146	66
6.1.1	For Signavio Process Manager integration	66
6.2	Updating to Signavio Workflow Accelerator 3.129	66
6.2.1	For Signavio Process Manager integration	67

6.2.2	Updating Node.js	67
6.2.3	New Mongo DB version supported	67
6.2.4	Updating the configuration for Single Sign-On using Windows and Kerberos/SPNEGO	68
6.3	Updating to Signavio Workflow Accelerator 3.99	68
6.3.1	For Signavio Process Manager integration	69
6.3.2	New Mongo DB version supported	69
6.4	Updating to Signavio Workflow Accelerator 3.85	69
6.4.1	For Signavio Process Manager integration	69
6.4.2	Data format of data sent from connectors	69
6.4.3	Updating Node.js	70
6.5	Updating to Signavio Workflow Accelerator 3.72	70
6.6	Updating to Signavio Workflow Accelerator 3.56	71
6.6.1	Updating the configuration file	71
6.6.2	Updating Node.js	72
6.7	Updating to Signavio Workflow Accelerator 3.39	73
6.8	Updating to Signavio Workflow Accelerator 3.22	73
6.8.1	Updating MongoDB	73
6.9	Updating to Signavio Workflow 3.15	75
6.10	Updating to Signavio Workflow 3.11	75
6.11	Updating to Effektiv 3.4	76
6.11.1	Updating Java and Tomcat	76
6.11.2	Updating Effektiv	77
6.11.3	Updating MongoDB	77
7	Starting Signavio Workflow Accelerator.....	80
7.1	Checklist.....	80
7.2	Start Workflow Accelerator	81
7.2.1	Windows	81
7.2.2	Script Engine	83
7.2.3	Mail Relay	83
7.2.4	Debian	83

7.3	Logging in to Workflow Accelerator	85
7.4	Create the initial user	86
7.5	Invite other users	87
8	Stopping Signavio Workflow Accelerator.....	88
8.1	Windows	88
8.1.1	Tomcat	88
8.1.2	MongoDB	88
8.1.3	Script Engine	88
8.1.4	Mail Relay	89
8.2	Debian	89
8.2.1	Tomcat	89
8.2.2	MongoDB	89
8.2.3	Script Engine	89
8.2.4	Mail Relay	90
9	Backup.....	91
9.1	Windows	92
9.2	Debian	92
10	Restoring backups.....	93
10.1	Windows	93
10.2	Debian	93
11	Tooling.....	94
11.1	MongoDB	94
11.1.1	Connecting to MongoDB using Robo 3T	94
11.1.2	Executing a JavaScript File for MongoDB	101
12	Troubleshooting.....	103
12.1	Where are the log files?.....	103
12.1.1	Workflow Accelerator	103
12.1.2	Script Engine	103

12.1.3 Mail Relay	104
12.1.4 MongoDB	104
12.1.5 Tomcat	104
12.2 Common problems	105

1 Introduction

This is the Signavio Workflow Accelerator administration manual. It will provide you with information to setup the Workflow Accelerator system on your own server infrastructure. Workflow Accelerator officially supports Microsoft Windows Server 64Bit and Debian Stable 64Bit.

It is possible to setup Workflow Accelerator also on other Linux distributions. However, the instructions in this manual are only tested for Debian. Probably, you will have to adjust the instructions to your system, but the used software should work on most distributions.

1.1 Software components

In order to setup and work with Workflow Accelerator, several software components are necessary.

1.1.1 Workflow Accelerator system

The Workflow Accelerator system refers to the application that you will run on your server and that consists of a Java web archive (WAR) that is deployed on Tomcat and a MongoDB database.

1.1.2 Apache Tomcat

The Apache Tomcat is an application server that hosts the Workflow Accelerator web application. The Workflow Accelerator web application serves the Workflow Accelerator application to the users' browser and it also includes the backend of the Workflow Accelerator system.

1.1.3 MongoDB

MongoDB is a document-oriented database system. It is the only database system that Workflow Accelerator supports. You can find more detailed information in the [MongoDB documentation](#).

1.1.4 Node.js

Node.js is a runtime environment for JavaScript which is used by Workflow Accelerator to execute custom JavaScript tasks created by the user.

1.1.5 Mail server (SMTP)

The mail server is necessary to send notifications and invitation requests to users. Any server that is accessible via SMTP can be used.

1.1.6 Web browser

A current web browser is necessary on every client that should display the Workflow Accelerator user interface.

See <https://www.signavio.com/browser-compatibility/> for a list of supported browsers.

1.2 Naming conventions

Within this administrator's guide, several placeholders are used to increase readability.

<code>\$WORKFLOW_HOME</code>	Refers to the directory that contains the unzipped content of the downloaded <code>signavio-workflow-{version}.zip</code> .
<code>\$TOMCAT_HOME</code>	Refers to the directory where your Apache Tomcat is installed. This directory contains the folders <code>webapps</code> and <code>conf</code> . On Linux for example the directory might be located in <code>/var/lib/tomcat8</code> .
<code>\$MONGO_HOME</code>	Refers to the directory where your MongoDB is installed.
<code>\$SCRIPT_ENGINE_HOME</code>	Refers to the directory where your Workflow Accelerator script engine is installed.
<code>\$MAIL_RELAY_HOME</code>	Refers to the directory where your Workflow Accelerator mail relay is installed.

Those placeholders are used within example command line code. You need to replace those placeholders with the appropriate paths on your system.

2 Installing Signavio Workflow Accelerator

This section will guide you through the set-up of Workflow Accelerator. Simply follow the subsections step by step. If you have already installed Workflow Accelerator and want to update the system to a newer version, you can skip this chapter.

2.1 Unpack Signavio Workflow Accelerator

Unzip `signavio-workflow-{version}.zip` to a directory of your choice, e.g. `/yourlocation/signavio-workflow-{version}`. In the rest of this guide, we'll refer to the Workflow Accelerator home directory as `$WORKFLOW_HOME`. The directory will contain the following files:

- *.sh - Linux shell scripts
- db - directory that contains MongoDB configuration files
 - mongod.conf - configuration file for Linux
 - mongod.cfg - configuration file for Windows
- mail-relay - directory that contains the mail relay
- script-engine - directory that contains the script engine
- optional - directory that contains optional resources
 - spnego - directory that contains resources to setup SSO with SPNEGO
- webapps - directory
- ROOT - this directory contains the web application

On Linux systems you need to make the scripts executable with the `chmod` command like this:

```
chmod a+x $WORKFLOW_HOME/*.sh
```

2.2 System requirements

2.2.1 Operating system

The Workflow Accelerator system can be deployed on different operating systems. Officially supported are the following operating systems:

2 Installing SignavioWorkflow Accelerator

- Debian Stable Release 64-bit
- Microsoft Windows Server 64-bit

In general, it is possible to install Workflow Accelerator on other Linux distributions. However, the installation process can differ from the following instructions and the Signavio team cannot offer official support in case of an installation problem.

You are advised to use a 64-bit operating system in order to be able to assign more than 1.5 GB of heap space to the Java VM. Furthermore, MongoDB is limited to 2 GB of content on 32-bit operating systems.

2.2.2 Hardware requirements

The described installation in this administrator's guide will set up the database and the application server on the same system. However, it is also possible to set up the database and application server on separate systems. For instance, if you already have an infrastructure that provides MongoDB instances for different applications, you can simply reuse your infrastructure and just create another database for Workflow Accelerator.

The minimal hardware requirements for the combined system of application and database server are:

- 8 GB RAM
- 4 core CPU
- 50 GB disk space

The required disk space depends on how you will use Workflow Accelerator. 50 GB should be sufficient disk space for most scenarios. However, if you plan on using a lot of file uploads in your processes the required disk space depends largely on the number and size of the uploaded files.

Note that the required disk space does not include the space necessary for backups. If you plan on storing the backups on the same server, you should add more.

2.2.3 Other systems

The Workflow Accelerator system relies on a SMTP mail server to send notifications and invite users. You can simply reuse any existing mail server in your organization that supports access via SMTP.

2.2.4 Software requirements

In order to run Workflow Accelerator on your server, you will need to install some additional software. Depending on your deployment strategy (combined or separate system for application and database), you have to install the following required software either on the same server or the respective separate servers. The following sections contain detailed instructions on how to install and configure the additional software components.

The application server requires the following additional software:

- Java: AdoptOpenJDK 8 or 11 (LTS). We recommend to use AdoptOpenJDK Version 8 (LTS), with Hotspot selected as the JVM.

Pay attention to the version number. Signavio Workflow Accelerator does not yet support OpenJDK 14.

- Apache Tomcat 8.5 64-bit

The database server requires the following additional software:

MongoDB version (version 3.4.x, 3.6.x, 4.0.x, or 4.2.x Workflow Accelerator does not support MongoDB version 4.4.x.)

The optional script engine requires the following additional software:

Node.js 12.x LTS

2.2.5 Web browser

A current web browser is necessary on every client that should display the Workflow Accelerator user interface.

See <https://www.signavio.com/browser-compatibility/> for a list of supported browsers.

2.3 Installing Java

The application server requires a Java runtime environment in order to execute Workflow Accelerator's Java backend. You must install Java 8 64-bit or Java 11 64-bit on the application server. Download an installation package for your operating system here:

<https://www.java.com/en/download/manual.jsp>

2.4 Installing Tomcat

The Java backend of the Workflow Accelerator system is executed using Apache Tomcat. Workflow Accelerator supports Apache Tomcat 64-bit version 8.5. It does not support Apache Tomcat 64-bit 9.

You can download the latest installation files for different operating systems from the following site:

<https://tomcat.apache.org/download-80.cgi>

In general, please make sure the user executing the Tomcat has read/write permissions for the `$TOMCAT_HOME/webapps` directory and its sub-directories.

2.4.1 Windows

If you are using a Windows system, the download site offers a Windows Service Installer (*32-bit/64-bit Windows Service Installer*). This installer will install the Apache Tomcat as a Windows service which can be started, stopped and configured over a graphical user interface. The service installer is the recommended way for Windows users to install the Tomcat server. Make sure you execute the installer with administrator privileges.

Furthermore, it is recommended to install the Apache Tomcat into a root directory on `C:\`.

2.4.2 Debian

If you are using Linux, consider installing Apache Tomcat using a package manager. For Debian 9, you can use the Advanced Packaging Tool (APT) and execute the following command:

```
apt-get install tomcat8
```

For older Debian versions, this command does not install Tomcat 8.5.

2.5 Configuring Tomcat

This section will point the configuration options that are required to run Workflow Accelerator. However, Tomcat offers more configuration options that can be found in the official documentation:

<https://tomcat.apache.org/tomcat-8.5-doc/>

2.5.1 Java options

There are some Java options that need to be configured before starting the Tomcat server. These options are configured using the `JAVA_OPTS` environment variable. It should contain the following values:

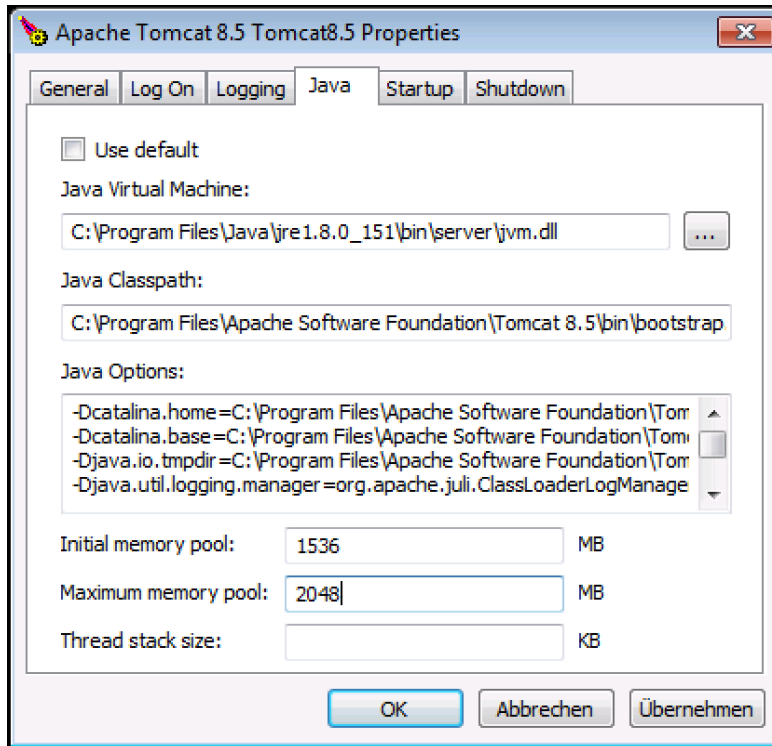
<code>-Xms1536m</code>	minimum RAM, should be at least 1.5GB
<code>-Xmx2048m</code>	maximum RAM, should be at least 2GB but more is recommended
<code>-Dfile.encoding=UTF-8</code>	sets UTF-8 as Apache Tomcat's default encoding mechanism
<code>-XX:PermSize=128m</code>	initial permanent generation space in MB
<code>-XX:MaxPermSize=256m</code>	maximum permanent generation space in MB

In order to allocate sufficient RAM to the Tomcat server you have to set the `Xms` and `Xmx` options, followed by the number of MB allocated and a literal `m`. `Xms` defines the initial/minimum amount of RAM that is used and `Xmx` the allowed maximum. You are advised to allocate at least 2GB of RAM to the Tomcat. However, it is recommended to use more RAM. For example, if you want to assign 4GB of maximum RAM the option would be `-Xmx4096m`.

The values for the permanent generation space have to be increased as well, these are defined by the values `-XX:PermSize=128m` and `-XX:MaxPermSize=256m`.

2.5.2 Windows

If you installed Tomcat using the Windows service installer, you can open the configuration dialog by opening the Windows start menu and selecting `Program Files\Apache Tomcat\Configure Tomcat`



The Java tab offers the possibility to set the options and define the memory pool (minimum and maximum RAM usage). Fill in the values for Xms and Xmx into the fields for memory pool and add the other options to the text area labeled as Java Options.

2.5.3 Debian

One way to define the `JAVA_OPTS` under Debian and other Linux distributions is to create the file `setenv.sh` in the Tomcat `/bin/` directory. This directory contains also other files like the `catalina.sh`, the `startup.sh` or the `shutdown.sh` and is for instance located in `/usr/share/tomcat8` or a similar directory. Create or extend any existing `setenv.sh` file and add the line for the `JAVA_OPTS`:

```
export JAVA_OPTS="$JAVA_OPTS -Xms1536m [... add the other options]"
```

In case you have created a new `setenv.sh` file, you have to make sure it is executable using the `chmod` command:

```
chmod a+x setenv.sh
```

2.5.4 Set up the Tomcat connector

By default the Tomcat server will start up using port 8080 and 8005. Port 8080 is used for serving the Workflow Accelerator web application. Port 8005 is used to shut down the Tomcat instance. You can change the port in the server.xml which is located in `$TOMCAT_HOME/conf/server.xml`.

If the server is running on any other port than port 80, your users will have to enter the port in the address line of the web browser, e.g.

```
http://workflow.yourdomain.com:8080/.
```

Locate the HTTP connector and change the value of port to your preferred port. Furthermore, make sure the connector contains the entry `URIEncoding="UTF-8"`. For example, the connector could look like this:

```
<Connector port="80" protocol="HTTP/1.1"  
  connectionTimeout="20000"  
  redirectPort="8443"  
  URIEncoding="UTF-8" />
```

2.6 Installing the Workflow Accelerator web application

The Workflow Accelerator web application, which can be found in the directory `$WORKFLOW_HOME/workflow/ROOT`, needs to be copied to the Tomcat's `webapps` directory. Remove any files and directories that are located in `$TOMCAT_HOME/webapps` and copy the `ROOT` directory to `$TOMCAT_HOME/webapps/`.

The Workflow Accelerator application needs to run as the `ROOT` application in the Tomcat. Running it in parallel with another application that requires to run as the `ROOT` application in the same Tomcat is currently not possible.

2.7 Installing MongoDB

MongoDB is currently the only database system that Workflow Accelerator supports. If you already have infrastructure that can provide a MongoDB instance (version 3.4.x, 3.6.x, 4.0.x, or 4.2.x Workflow Accelerator does not support MongoDB version 4.4.x.), create a new instance for Workflow Accelerator and skip to **Configuring MongoDB**. Otherwise continue with the installation instructions. General information about installing MongoDB on different operating systems can be found at:

<https://docs.mongodb.com/manual/installation/>

There are two versions of MongoDB, *MongoDB Community Edition* and *MongoDB Enterprise*. *MongoDB Community Edition* can be used free of charge even in a commercial context, whereas *MongoDB Enterprise* is a commercial product that offers additional tooling and support. You can use either version with Workflow Accelerator.

In case you are using Linux, you have the possibilities to install MongoDB using either a package manager like `apt-get` or downloading the binaries directly. It is recommended to use the package manager because it will also set up scripts that will allow you to easily start and stop the server. However, you have to ensure the correct version is installed.

Debian explains in more detail how to set up MongoDB on Debian, and might work as an example for other Linux distributions.

2.7.1 Remarks for downloading the binaries

You can find a version of MongoDB for your operating system here:

<https://www.mongodb.com/download-center>

2.7.2 Windows

Windows requires you to download the binaries. Consider the remarks of the last subsection when doing so.

You can find more details about installing MongoDB on Windows in the documentation:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>

When you have finished downloading MongoDB, follow these steps:

1. Start the installation by double-clicking the downloaded MongoDB `.msi` and following the instructions.
 - You can select another install directory, e.g. `C:\MongoDB`, if you choose the *Custom* installation option.
2. Create a data directory for the MongoDB files, e.g. `C:\MongoDB\data`.
 - This directory will need the most disk space. Make sure the drive has sufficient disk space.
3. Create a logs directory for the MongoDB log files, e.g. `C:\MongoDB\logs`.
4. Copy the file `$WORKFLOW_HOME\db\mongod.cfg` to your MongoDB directory `C:\MongoDB\mongod.cfg` and edit the file.
 - The `dbPath` under `storage` must contain the absolute path to the data directory, e.g. `C:\MongoDB\data`.
 - The `path` under `systemLog` must contain the absolute path to the log file, e.g. `C:\MongoDB\logs\mongodb.log`. The log file will be created once MongoDB is started.
 - The user executing MongoDB needs access to the data directory and the log file.
 - The configuration file needs to be in a valid **YAML** format. You can use an online checker like **YAML Lint** to verify the validity.
 - See **Configuring MongoDB** for more information about the other configuration values.

You no longer need to carry out the following 3 steps for MongoDB 4.0 or higher. The installer registers the service automatically. You can find the service in `services.msc`.

5. Open the command line `cmd` with administrative privileges and execute the following command. Make sure to use absolute paths and replace them with the matching ones on your system.
 - `C:\MongoDB\bin\mongod.exe --config C:\MongoDB\mongod.cfg --install`
 - The command line should tell you that the service MongoDB was properly created.

2 Installing SignavioWorkflow Accelerator

- You can unregister the service again by executing:

```
C:\MongoDB\bin\mongod.exe --remove
```

- The registered service should start MongoDB automatically on startup of Windows.

6. You can start MongoDB by executing the following command:

```
net start MongoDB
```

7. You can stop MongoDB by calling:

```
net stop MongoDB
```

- The service is also listed in the services window that can be opened by running

```
services.msc
```

2.7.3 Debian

If you are installing MongoDB on a Debian system, we recommend to use the method described in the MongoDB configuration:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/>

You can now start and stop the MongoDB server by calling :

```
sudo service mongod start
```

or

```
sudo service mongod stop
```

The server will use the default configuration file `/etc/mongod.conf`. See the **Configuring MongoDB** section for more information on how to configure MongoDB for Workflow Accelerator and create the required user. Note that the installation probably already started the MongoDB server. You need to restart the MongoDB server after you edit the configuration.

2.8 Configuring MongoDB

In order to run MongoDB properly, some configuration options have to be defined. The easiest solution is to create a configuration file and link this configuration file when starting MongoDB with the `--config` option. The MongoDB

configuration uses the **YAML** format, you can use an online checker like **YAML Lint** to verify that your configuration file has the proper format.

If you installed MongoDB under Windows using the method described in **Installing MongoDB**, you have already copied the predefined configuration file and adjusted the values for `dbPath` and `systemLog path` to your system. Then your configuration file already contains all necessary values.

If you installed MongoDB under Linux by following the instructions in **Installing MongoDB** using `apt-get` there should be a `/etc/mongod.conf` configuration file. Edit this configuration file. The `dbPath`, `systemLog path` properties should already be defined. We recommend that you set the `logAppend` property to `true` and the `authorization` property to `enabled`.

In any other case make sure the configuration file contains the following properties:

```
systemLog:
  destination: file
  path: /var/log/mongodbdb/mongoddb.log
  logAppend: true
storage:
  dbPath: /var/lib/mongodb
security:
  authorization: enabled
```

<code>dbPath</code>	Defines the directory where the database files are stored.
<code>path</code>	Defines the file that will contain the logging output.
<code>logAppend</code>	Indicates that new logs will be appended to an existing log file after restarting the server, if set to true.
<code>authorization</code>	(optional) Turns authentication on, if set to <code>enabled</code> . It is advised to turn authentication on in case the MongoDB instance can be accessed over the network.

If you have installed MongoDB on Debian using a package manager, the default configuration file will probably contain the following entry: `net: bindIp: 127.0.0.1` This setting tells MongoDB to only bind to the local interface and reject remote connections. If you plan to install MongoDB and Tomcat on different machines, you must update this setting and either remove it or add the IP interface MongoDB should bind to, in order to listen for incoming connections. This value can contain a comma separated list of IPs and should NOT contain the IP address of the application server.

For more configuration options, see

<https://docs.mongodb.com/manual/reference/configuration-options/>.

2.8.1 Add a database user for Workflow Accelerator

You have two options for setting up the database, see the following sections **Setting up the database for the first time** or **Setting up the database as an advanced user**, here you add a DB user with limited access.

In order to add a new user to MongoDB, the MongoDB server has to be running.

2.8.1.1 Setting up the database for the first time

Workflow Accelerator requires a MongoDB user in the admin database that has the following roles:

<code>dbAdminAnyDatabase</code>	
<code>readWriteAnyDatabase</code>	
<code>clusterAdmin</code>	This role will allow you to list all databases. This becomes necessary if you plan on using the user credentials when editing any database content with a tool like Robomongo. Furthermore, it is necessary for creating backups.
<code>userAdminAnyDatabase</code>	This role is required for backups.

The user will access the databases for Workflow Accelerator. Workflow Accelerator will create two databases, one for the user and workflow data, and one for uploaded files. The name of the first database can be configured in the Workflow

Accelerator configuration file. The name of the second database is derived from the name of the first one by adding “-files”. For instance, if you define the database name “signavio”, the databases “signavio” and “signavio-files” will be created.

In general, if you want to create a new user in MongoDB you will need to authenticate with an existing user that has the role `userAdmin` or `userAdminAnyDatabase`. There is an exception for a fresh MongoDB set-up. It will allow you to create the first user from localhost without any authentication. The following examples will show you how to create a new user using the credentials of an admin user and how to do it without any credentials in case of a new MongoDB.

To set up a new MongoDB server for the first time, follow the procedure for your operating system.

Windows

1. Open the command line `cmd` and go to your MongoDBbin directory, e.g. by executing:

```
cd C:\MongoDB\bin
```

2. If you have an admin user, create the Workflow Accelerator user by executing:

3.

```
mongo.exe admin -u admin -p  
<enterYourAdminPasswordHere> --eval "db.createUser( {  
user: 'signavio', pwd: '<enterSignavioPasswordHere>',  
roles: ['dbAdminAnyDatabase', 'readWriteAnyDatabase',  
'clusterAdmin', 'userAdminAnyDatabase'] } )"
```

- Replace `<enterAdminPasswordHere>` with the password you have defined for the admin user.
 - Replace `<enterSignavioPasswordHere>` with the password for the new Workflow Accelerator user. As you can see the user name is `signavio`.
4. If you have a new MongoDB and no admin user, create the Workflow Accelerator user by executing:

2 Installing SignavioWorkflow Accelerator

```
mongo.exe admin --eval "db.createUser( { user:
'signavio', pwd: '<enterSignavioPasswordHere>',
roles: ['dbAdminAnyDatabase', 'readWriteAnyDatabase',
'clusterAdmin', 'userAdminAnyDatabase'] } )"
```

- Replace <enterSignavioPasswordHere> with the password for the new Workflow Accelerator user. As you can see the user name is signavio.

5. You can verify the creation of the new Workflow Accelerator user by executing:

```
mongo.exe admin -u signavio -p
<enterSignavioPasswordHere> --eval "db.getUser
('signavio');"
```

- On the command line you will see the user information including the four assigned roles.

Debian

If you have installed MongoDB using a package manager, the binaries should be available on the path. Otherwise, open the directory with the MongoDB binary files and follow the instructions.

1. If you have an admin user, create the Workflow Accelerator user by executing:

```
mongo admin -u admin -p <enterYourAdminPasswordHere>
--eval "db.createUser( { user: 'signavio', pwd:
'<enterSignavioPasswordHere>', roles:
['dbAdminAnyDatabase', 'readWriteAnyDatabase',
'clusterAdmin', 'userAdminAnyDatabase'] } )"
```

- Replace <enterAdminPasswordHere> with the password, you have defined for the admin user.

2 Installing SignavioWorkflow Accelerator

- Replace `<enterSignavioPasswordHere>` with the password for the new Workflow Accelerator user. As you can see the user name is `signavio`.
2. If you have a new MongoDB and no admin user, create the Workflow Accelerator user by executing:

```
mongo admin --eval "db.createUser( { user:
'signavio', pwd: '<enterSignavioPasswordHere>',
roles: ['dbAdminAnyDatabase', 'readWriteAnyDatabase',
'clusterAdmin', 'userAdminAnyDatabase'] } )"
```

- Replace `<enterSignavioPasswordHere>` with the password for the new Workflow Accelerator user. As you can see the user name is `signavio`.
3. You can verify the creation of the new Workflow Accelerator user by executing:

```
mongo admin -u signavio -p
<enterSignavioPasswordHere> --eval "db.getUser
('signavio');"
```

- On the command line you will see the user information including the four assigned roles.

Once the Workflow Accelerator user is created, its credentials have to be added to the Workflow Accelerator configuration file to the `effektif.mongodb.username` and `effektif.mongodb.password` properties. Section **Update the Workflow Accelerator configuration file** explains how to update the configuration file.

2.8.1.2 Setting up the database as an advanced user

If you already have a MongoDB cluster, you can add the required databases manually and add a new database user with limited access to the two required databases for Workflow Accelerator.

You can define the name for the first database which stores workflow documents. The name for the second database is derived from the first name by adding the suffix `-files`. If you name your first database `signavio`, the second database must be named `signavio-files`.

The database user should have the following built-in role for both databases:

- readWrite

The following mongo shell example adds a user to the admin database that has the required access to the databases `signavio` and `signavio-files`. Make sure to change `user`, `pwd` and the database names to your respective values.

```
use admin
db.createUser({
  user: 'signavioUser',
  pwd: 'secret',
  roles: [
    { role: 'readWrite', db: 'signavio' },
    { role: 'readWrite', db: 'signavio-files' },
  ]
})
```

Once the Workflow Accelerator user is created, its credentials have to be added to the Workflow Accelerator configuration file to the `effektif.mongodb.username` and `effektif.mongodb.password` properties. Section **Update the Workflow Accelerator configuration file** explains how to update the configuration file.

2.8.2 SSL setup

The following sections describe how you have to change the configuration of the web application to enable SSL/TLS transport encryption for the database connection.

2.8.2.1 Enable SSL / TLS transport encryption for MongoDB

To enable SSL/TLS transport encryption on MongoDB connection for Workflow Accelerator, the configuration file `effektif-onpremise.properties` requires the entry `effektif.mongodb.uri` with a proper MongoDB URI value and the parameter `ssl` set to true. See section **Update the Workflow Accelerator configuration file** for more information about the `effektif-onpremise.properties` file and where to find it.

1. Open the configuration file `effektif-onpremise.properties`.
2. Add the property `effektif.mongodb.uri` and a MongoDB connection URI as the value.

2 Installing SignavioWorkflow Accelerator

- The URI can include several options, [see MongoDB documentation for details](#).
- Include the option `ssl=true` to enable SSL / TLS.
- The URI has to specify the database used by Workflow Accelerator.

3. Start or restart the Tomcat application server.

Workflow Accelerator uses two databases, one for workflows and cases and the second one for contents of uploaded files. The name of the second database is derived from the name of the first database by adding the suffix `-files`. Following the example above, the name of the second database would be `signavio-files`. Ensure the database user for Workflow Accelerator has access to the second database as well.

2.8.2.2 Create a MongoDB connection URI

A MongoDB connection URI specifies to which servers and database a client should connect. The URI includes authentication credentials as well as additional options.

A typical URI consists of the following parts:

- Database user name
- Database user password
- Hostname of the database
- Port of the database - default value is 27017
- Name of the database

Structure: `mon-`

`god-`

`b://<DBUser-`

`Name>:<DBUserPassword>@<Hostname>:<Port>/<DatabaseName>?authSource=admin`

The last part `?authSource=admin` specifies the first option authenticate to the admin database. Usually users are created in that database.

Example

An example value for a MongoDB connection URI with enabled SSL option looks like this: `effek-`

`tif.-`

`mon-`

`god-`

2 Installing SignavioWorkflow Accelerator

```
b.ur-  
i=mo-  
ngod-  
b://myUser-  
:myPassword@mongodb01.example.com:27017/signavio?authSource=admin&ssl=true
```

The example connects to the database `signavio` on the server `mon-
godb01.example.com` using `myUser` and `myPassword` as credentials. SSL / TLS is turned on and the authentication database is set to `admin`.

As an alternative you can use the **DNS seedlist connection format**. Use a connection string prefix of `mongodb+srv` instead of `mongodb` to enable it.

2.8.2.3 Migrating from old to new database configuration

Four properties were previously used to configure the MongoDB connection. These properties are obsolete and will be ignored once the new MongoDB URI property is configured.

- o `effektif.mongodb.servers`
- o `effektif.mongodb.username`
- o `effektif.mongodb.password`
- o `effektif.mongodb.database`

If you have previously used these configuration parameters, you can use the values to create the new MongoDB URI value.

Example

With this example configuration

- o `effektif.mongodb.servers=mongodb01.example.com`
- o `effektif.mongodb.username=myUser`
- o `effektif.mongodb.password=myPassword`
- o `effektif.mongodb.database=signavio`

the new MongoDB URI configuration would look like this:

```
effek-  
tif.-  
mon-  
god-  
b.ur-  
i=mo-  
ngodb://myUser:myPassword@mongodb01.example.com:27017/signavio?authSource=admin
```

Once you have configured the new MongoDB URI property, you can safely remove the old database configuration properties from the configuration file.

2.8.2.4 Using self signed certificates

When you use a self signed certificate in your MongoDB SSL / TLS configuration created by your own certificate authority, you need to configure the JVM/Tomcat server to trust your certificate authority. Otherwise the certificate validation will fail and Workflow Accelerator will not be able to create a connection to MongoDB.

To create a trust store for your root CA certificate and configure it in Tomcat, follow these steps:

1. Follow the instructions in the official MongoDB documentation to create a Java trust store for your custom CA certificate chain.
 - Name the trust store (in this example `ca_cert.jks`)
 - Set a password for the trust store (in this example `caPassword`)
2. Place the trust store in a location which can be accessed by the Tomcat that runs Workflow Accelerator.
 - Ensure the Tomcat user has access to the trust store file and can read it.
 - In this example we use `/certs/ca_cert.jks`.
3. Extend the configured Java options in Tomcat to use the new trust store. Add the following environment variables (replace the values with your respective values)
 - `Djavax.net.ssl.trustStore=/certs/ca_cert.jks`
 - `Djavax.net.ssl.trustStorePassword=caPassword`
 - `Djavax.net.ssl.trustStoreType=JKS`
4. Restart Tomcat.

Now Workflow Accelerator is able to connect to the MongoDB server and verify its identity.

2.8.3 Backup and restore

You are advised to do regular backups of the MongoDB database to prevent a complete data loss in case of a system failure. The Workflow Accelerator system will NOT backup the user data on its own. **Backup** explains in detail how to set up backups for MongoDB properly. **Restoring backups** explains how you can restore an older version of the user data from a backup.

2.9 Installing Node.js

You only need to install and configure Node.js if you purchased a version of Workflow Accelerator which allows you to use JavaScript tasks.

Node.js is a runtime environment for JavaScript which is used by Workflow Accelerator to execute custom JavaScript tasks created by the user. We require Node.js LTS (Long Term Support) version 12.x.

2.9.1 Windows

For Windows, there is a self-extracting installer to set up Node.js.

1. Go to <https://nodejs.org/en/download/>
2. Select the LTS version.
3. Download the *Windows Installer (.msi)* for your Windows version.
4. Execute the downloaded installer and follow installation guide.

2.9.2 Debian

The Node.js site offers additional documentation for setting up Node.js on Debian using the package manager APT:

<https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions>

Make sure to install the LTS version of Node.js.

2.10 Installing the Workflow Accelerator script engine

You only need to install and configure the script engine if you purchased a version of Workflow Accelerator which allows you to use JavaScript tasks.

The Workflow Accelerator script engine uses Node.js to execute JavaScript tasks within a workflow and an embedded HTTP server to communicate with the Workflow Accelerator web application. You can install the script engine on either the same machine as the Workflow Accelerator web application or on a separate machine. However, if you use a separate machine to run the script engine, you have to make sure the web application can connect over HTTP to the configured port of the script engine and vice versa. In order to install the script engine, follow these instructions:

1. Create new local directory for the script engine,
e.g. `C:\Program Files\Script Engine` or `/var/lib/script-engine`.
 - We will refer to this directory as `SCRIPT_ENGINE_HOME`.
2. Copy the content of the directory `WORKFLOW_HOME/script-engine/` to your newly created directory.
 - You should find the file `server.js` directly within your directory: `SCRIPT_ENGINE_HOME/server.js`.

Next, edit two configuration files.

3. In the script engine configuration file `SCRIPT_ENGINE_HOME/configuration.onpremise.js` edit the values for the port, the log files and the Workflow Accelerator URL.
 - Ensure that the port for the script engine is not used by any other application.

<code>port</code>	Defines the port the script engine will bind to and listen for incoming HTTP requests.
<code>log file</code>	Defines the location and name of the script engine log file. You can use an absolute or relative path. By default, the log files are stored in the same directory as the script engine.
<code>log errorFile</code>	Defines the location and name of the script engine error log file.
<code>workflowUrl</code>	Defines the URL of the Workflow Accelerator system (including port), e.g. <code>http://localhost:8080</code>

4. In the the Workflow Accelerator configuration file `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/effektif-onpremise.properties`, add the URL of the script engine.
 - o This URL is not listed in the configuration file template, you need to add the complete property.
 - o If you run the script engine on the same machine as the web application and use the default port, the URL will be `http://localhost:8081`.

<code>effektif.javascript.server.url</code>	(Optional) The URL of the Workflow Accelerator script engine, e.g. <code>http://localhost:8081</code>
---	---

After the successful set-up you can start the script engine by opening the `SCRIPT_ENGINE_HOME` directory on a command line and executing the following command:

```
node server.js
```

2.10.1 Wrapping the script engine into a service

The script engine can be executed as shown above by running the command on a command line. However, it might be useful for production systems to run the script engine as a service or daemon. The following sub sections describe one possibility for each supported operating system which can be used to accomplish this goal. Note that Signavio offers no support for any third-party tools. Use them at your own discretion.

2.10.1.1 Windows

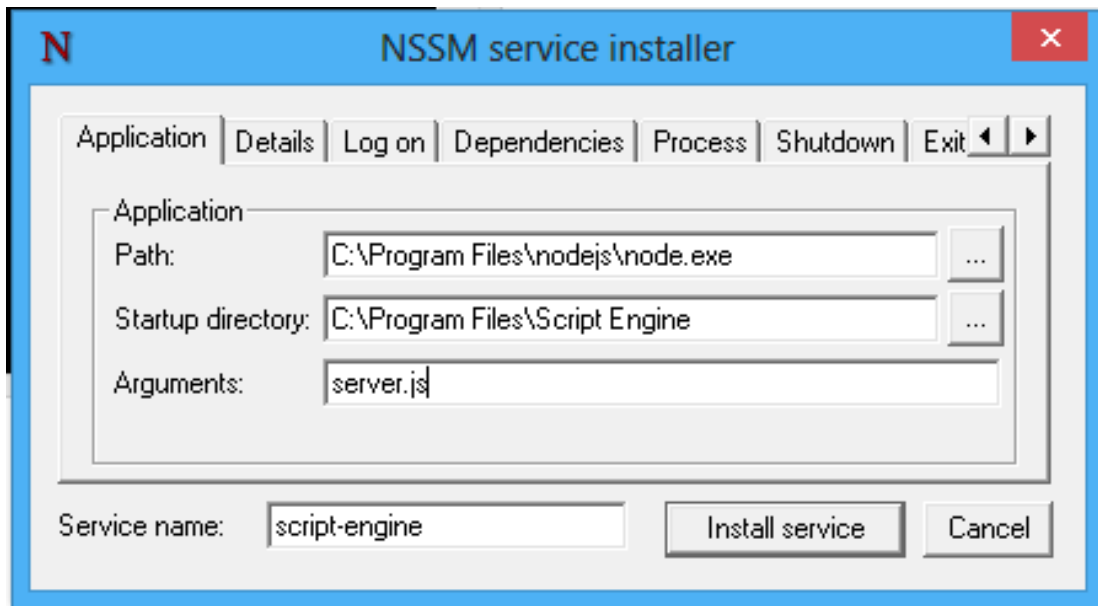
For Windows, there are several software tools which allow you to wrap a script into a service and monitor the execution. The **Non-Sucking Service Manager** (NSSM) is such a free tool which allows you to wrap any executable into a Windows service.

1. Download NSSM from <http://nssm.cc/download>
2. Create a new local installation directory for NSSM, e.g. `C:\Program Files\NSSM`.
3. Unzip the downloaded file and copy either the 64-bit `win64\nssm.exe` or the 32-bit `win32\nssm.exe` file to your new local directory.
4. Add the installation directory of NSSM to the environment variable `PATH` in the Windows system settings.
 - This is necessary to use the `nssm` executable on the command line without an absolute path.
5. Open the command line `cmd` with administrative privileges.
6. Execute

```
nssm install script-engine
```

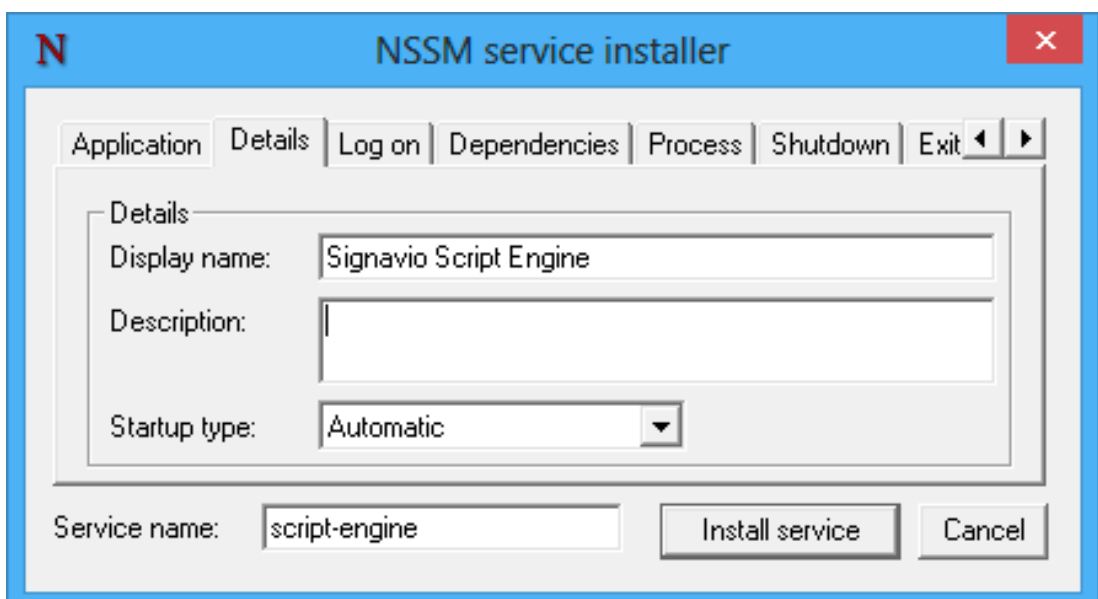
7. The NSSM configuration dialog opens
8. In the *Application* tab fill in the following information:

<i>Path</i>	the path to the Node.js .exe file, e.g. <code>C:\Program Files\nodejs\node.exe</code>
<i>Startup directory</i>	the value of <code>\$SCRIPT_ENGINE_HOME</code> , e.g. <code>C:\Program Files\Script Engine</code> .
<i>Arguments</i>	the exact value <code>server.js</code>



9. In the *Details* tab fill in the following information:

<i>Display name</i>	the name of the service shown in <code>services.msc</code>
<i>Description</i>	a description which will help you to recognize the service
<i>Startup type</i>	choose if the service shall start automatically on startup or if you want to start it manually



10. Fill in any other service configuration settings you deem appropriate for your system in the other tabs.
11. Click on *Install service*.

Now you can start the script engine with the following command:


```
nssm start script-engine
```

You can check the status of the script engine by executing:

```
nssm status script-engine
```

In order to stop the script engine again, execute the following command:

```
nssm stop script-engine
```

While it is running, you can open your browser and go to the URL you have configured for the script engine, by default this `http://localhost:8081`. If the engine is running, you will see a blank page with this content: 

Besides using `nssm` on the command line, you will find a new service when you open the `services.msc`. The script engine will show up under the name you configured as display name. You can start and stop the script engine from here as well.

2.10.1.2 Linux

On Linux this task is usually accomplished with Init scripts using the respective Init system of your Linux distribution. Because there are several different Init systems, Signavio will not provide a template. However, if you don't already have a template, you can find lots of matching templates for your Init system on the internet.

As mentioned before, the script engine can be started from within `SCRIPT_ENGINE_HOME` by executing:

```
node server.js
```

You have to ensure your Init script starts the Node.js executable `node` and uses the JavaScript file `SCRIPT_ENGINE_HOME/server.js` as an argument. Note, if you use a restricted user to execute the command, make sure the user has appropriate access to the `SCRIPT_ENGINE_HOME` directory.

2.11 Installing the Workflow Accelerator mail relay

You only need to install and configure the mail relay if you purchased a version of Workflow Accelerator which allows you to use email triggers.

The mail relay is a SMTP server which receives emails and forwards them via HTTP to the Workflow Accelerator web application. It is required in order to start a workflow by email. If you don't plan on starting workflows by email, you don't have to set this component up and can skip this section.

The complete set-up of the mail relay consists of the following steps:

1. Create an email domain for the mail relay
2. Install and configure the mail relay
3. Configure the web application

2.11.1 Create an email domain for the mail relay

In order to work properly, the mail relay needs to receive all relevant emails. The used email addresses are created dynamically based on the ID of the workflow that should be started. The domain used for the email addresses can be chosen by you and has to be configured in the system. We advise that you create a specific subdomain which is different from your normal email domain and redirect any incoming emails to the mail relay.

For example if you use the email domain `mail.yourcompany.com`, a typical email address for the workflow system would look like this:

```
process-5702854fd1dfff250dc57994@mail.yourcompany.com
```

How you route the emails to the mail relay is up to you and depends on your infrastructure. If the server which runs the mail relay is publicly available, you can simply set up a respective **MX record**. Note, the mail relay and the Workflow Accelerator web application don't have to run on the same server. If your email server (e.g. Microsoft Exchange) allows rerouting traffic for a specific subdomain to another server, you can also set up a respective rule there. Make sure the email text and sender address have not been modified due to some forwarding mechanism.

The next subsection explains how you set up the mail relay and configure it to use your email domain.

2.11.2 Install and configure the mail relay

The mail relay comes as a JAR file which requires Java 8 or 11 for the execution. Make sure Java is installed before continuing with the set-up.

The mail relay can be installed on the same server as the Workflow Accelerator web application or on a different server. If you choose two different servers, ensure the mail relay is able to reach the server which runs the web application.

1. Create a new local directory for the mail relay, e.g. `C:\Program Files\Mail Relay` or `/var/lib/mail-relay`.
 - We will refer to this directory as `$MAIL_RELAY_HOME`.
2. Copy the contents of the directory `$WORKFLOW_HOME/mail-relay/` to your newly created directory.
 - You should find the file `mail-relay.jar` directly within your directory: `$MAIL_RELAY_HOME/mail-relay.jar`
3. Within the `$MAIL_RELAY_HOME` directory create a new directory `logs`
4. Open `$MAIL_RELAY_HOME/logback.xml` and edit `value` in the following line:

```
<property name="LOG_DIR" value="."/>
```
5. Set `value` to the absolute path of `$MAIL_RELAY_HOME/logs`.
 - For example `C:\Program Files\Mail Relay\logs` or `/var/lib/mail-relay/logs`
 - On Windows you have to use `\` as a separator for the path.

Next, edit the configuration file.

6. Open `$MAIL_RELAY_HOME/mail-relay.properties` and edit the values for the domain, port and Workflow Accelerator URL.
 - Ensure that the port for the mail relay is not used by any other application.

The configuration file offers the following options:

<code>workflow.relay.domain</code>	Defines the (sub)domain which is used to receive emails for Workflow Accelerator.
<code>workflow.relay.port</code>	Defines the port the SMTP will use and listen to.
<code>workflow.relay.serverUrl</code>	Defines the URL of the Workflow Accelerator web application.

After the successful set-up you can start the mail relay by opening the command line and executing the following command :

```
java -jar $MAIL_RELAY_HOME/mail-relay.jar
```

Replace `$MAIL_RELAY_HOME` with the absolute path to the JAR file.

You can also register a new service or set up a new Linux Init script depending on your operating system to ease the management of the mail relay.

2.11.2.1 Windows

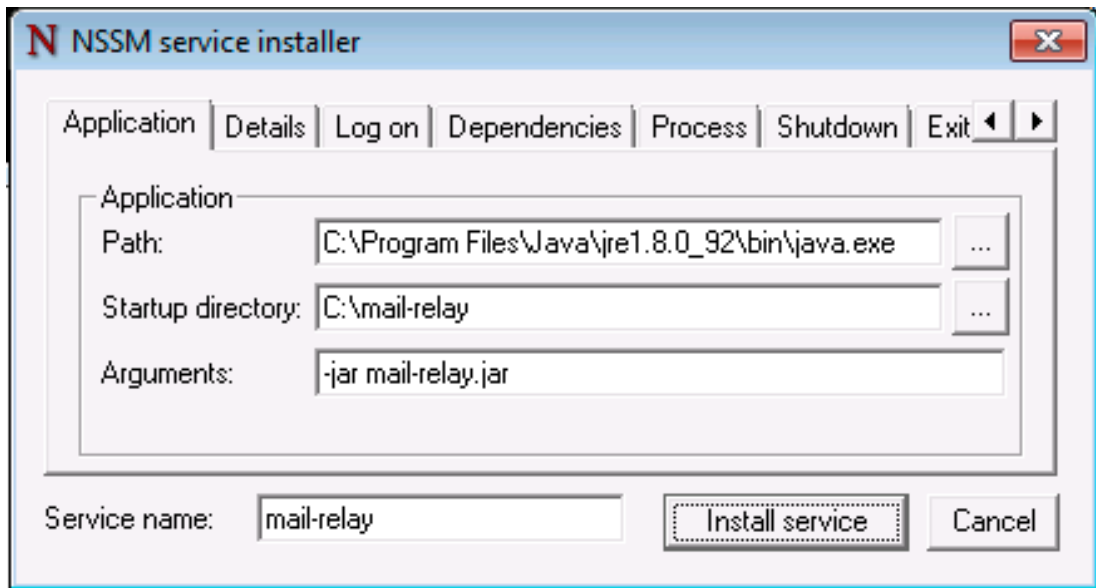
Section **Wrapping the script engine into a service** explains how you can wrap a single command into a service using NSSM. If you installed NSSM previously, you don't need to install it again. Simply follow the instructions and replace the details for the script engine with the details for the mail relay:

1. Open a command line with administrator privileges and execute:

```
nssm install mail-relay
```

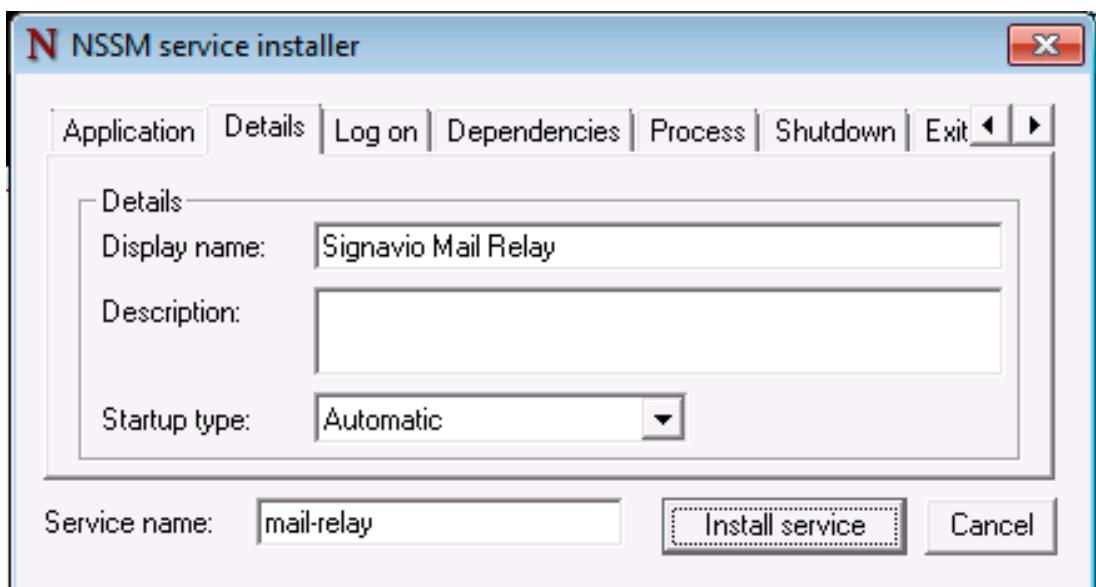
2. In the **Application** tab fill in the following information:

Path	the path to the Java .exe file, e.g. <code>C:\Program Files\Java\jre1.8.0_92\bin\java.exe</code>
Startup directory	the value of <code>\$MAIL_RELAY_HOME</code> , e.g. <code>C:\Program Files\Mail Relay</code> .
Arguments	the exact value <code>-jar mail-relay.jar</code>



3. In the **Details** tab fill in the following information:

Display name	the name of the service shown in <code>services.msc</code>
Description	a description which will help you to recognize the service
Startup type	choose if the service shall start automatically on startup or if you want to start it manually



After the successful registration, you can start the mail relay similarly to the script engine either on command line or via `services.msc`.

2.11.2.2 Linux

Similar to the script engine, for the mail relay this task is usually accomplished with Init scripts using the respective Init system of your Linux distribution.

Because there are several different Init systems, Signavio will not provide a template. However, if you don't already have a template, you can find lots of matching templates for your Init system on the internet.

As mentioned before, the mail relay can be started by executing:

```
java -jar $MAIL_RELAY_HOME/mail-relay.jar
```

The placeholder `$MAIL_RELAY_HOME` has to be replaced with the absolute path to the JAR file.

You have to ensure your Init script starts the Java executable and uses the arguments `-jar` and the absolute path to JAR file `mail-relay.jar`. Note that if you use a restricted user to execute the command, the user must have appropriate access to the `$MAIL_RELAY_HOME` directory.

2.11.3 Configure the web application

After setting up the mail relay, the email trigger needs to be enabled in the web application configuration. Section **Update the Workflow Accelerator configuration file** explains where you find the respective configuration file. Make sure to set proper values for the following configuration values:

```
effektif.mail.receiver.enabled = true  
effektif.mail.receiver.domain = mail.yourcompany.com
```

The first value must be set to `true` to enable the email trigger. The second value must be set to the email domain you created for the mail relay. It will be shown in the user interface of the web application.

2.11.4 Testing the mail relay

In order to verify the set-up works as intended, it can be helpful to test it. If you have installed Workflow Accelerator completely and created an organization within the application, you can directly test the set-up with a workflow.

2 Installing SignavioWorkflow Accelerator

1. Create a new workflow
2. Choose the email trigger
3. Copy the email address which is shown in the UI
4. Publish the workflow
5. Send an email to the copied email address

The case list should now contain a new completed case which shows your email in the event stream. If no case was started, you should check the log file `$MAIL_RELAY_HOME/logs/mail.log` for any error message.

If you haven't finished the complete set-up, but at least the web application is running as well, you can test already whether the communication between the mail relay and the web application works. A simple way to do that is to send an email via command line to the mail relay and consult the log file to see the result from the server. The following examples will show you how you can do that on different operating systems. The examples will send an email to a not existing workflow which is referenced by `process-123`. Therefore, the web application will respond with an error code and you will see a message similar to `Invalid response status code: 400` in the log file. This error is expected, as the workflow doesn't exist, and proves the communication is working.

2.11.4.1 Windows

You can send an email using the Windows PowerShell:>

```
Send-MailMessage -SMTPServer localhost -To process-123@mail.yourcompany.com -From you@yourcompany.com -Subject "This is a test email" -Body "This is the test message"
```

Replace the domain of the email address for the `-To` parameter with the one you set in the configuration file.

2.11.4.2 Linux

You can send an email using telnet to connect to the mail relay.

Open a command line on the server which is running the mail relay and execute:

```
telnet localhost 25
```

Replace the number `25` with the respective port that you configured for the mail relay.

Afterwards type in the following example line by line and replace the domain in the email for `rcpt to` to match the domain you configured for the mail relay.


```
helo me
mail from:<john.doe@yourcompany.com>
rcpt to:<process-123@mail.yourcompany.com>
data
From: john.doe@yourcompany.com
Subject: test subject

This is the body.
.
quit
```

2.12 Configuring Workflow Accelerator

2.12.1 Install the license file

You have received the Workflow Accelerator license file, `license.xml`, along with the Workflow Accelerator application. Before you can start the Workflow Accelerator system, you have to add the license file to the application. Copy the license file into the directory `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes`. The Workflow Accelerator system will check your license file on startup and set up the defined number of licenses.

2.12.2 Update the Workflow Accelerator configuration file

Before you can start Workflow Accelerator for the first time, you need to configure some mandatory properties in the Workflow Accelerator configuration file. Make sure you have copied the Workflow Accelerator web application to the Tomcat's webapps directory before continuing. The configuration file can be found here `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/effektif-onpremise.properties`.

The configuration file is a property file that contains one configuration option per line. Key and value are separated by a `=`. Every line that starts with a `#` is commented out and will not be used.

In general, the configuration allows to configure the base URL of the Workflow Accelerator system, the mail server, the database connection and integrations with third party systems (e.g. Signavio). If you installed the Workflow Accelerator script engine, you need to configure the URL to the script engine as well. As

2 Installing SignavioWorkflow Accelerator

described in **Installing the Workflow Accelerator script engine** the URL derives from the domain (`localhost` for the same machine) and port the script engine is running on. If you installed the Workflow Accelerator mail relay, you need to enable the email trigger and set the email domain that you created during the set-up.

2.12.2.1 All configuration options

<code>effektif.baseUrl</code>	(Required) The IP address or server name (including the port) of the server running the Workflow Accelerator Tomcat application server. E.g. <code>http://workflow.yourdomain.com:8080</code> If the server is running on port 80, the port can be omitted.
<code>effektif.smtp.host</code>	(Required) The IP address or server name of the outgoing email server.
<code>effektif.smtp.port</code>	The port on which the outgoing SMTP server listens
<code>effektif.smtp.ssl</code>	If SSL should be used (true or false)
<code>effektif.smtp.tls</code>	If TLS should be used (true or false)
<code>effektif.smtp.user</code>	The username for authentication
<code>effektif.smtp.password</code>	The password for authentication
<code>effektif.mail.from</code>	This email address is used as the sender when the Workflow Accelerator system sends out notifications.
<code>effektif.mail.feedback</code>	This email address will receive all feedback which is sent using the feedback functionality in the application.
<code>effektif.mongodb.servers</code>	A comma-separated list of MongoDB servers (hostnames). If you have MongoDB running on the same server as Tomcat, the default value of localhost is okay.
<code>effektif.mongodb.username</code>	The Workflow Accelerator MongoDB user name. If you created the user following the instructions in this guide, the name is <code>signavio</code> .
<code>effektif.mongodb.password</code>	The password of the Workflow Accelerator MongoDB user. This is the password you defined during user creation.
<code>effektif.mongodb.database</code>	The name of the database Workflow Accelerator should use. The default value <code>signavio</code> is okay.
<code>effektif.mongodb.uri</code>	(Optional) A MongoDB connection URI can be set as alternative to the database configuration properties above. Once set, the other database configuration properties are ignored (see section Create a MongoDB connection URI).
<code>effektif.javascript.server.url</code>	(Optional) The URL of the Workflow Accelerator script engine, e.g. <code>http://localhost:8081</code>
<code>effektif.mail.receiver.enabled</code>	(Optional) Activates the email trigger if set to <code>true</code> . You have to set up the mail relay for the trigger to work properly.
<code>effektif.mail.receiver.domain</code>	(Optional) The email domain which is used to receive emails for the email trigger. The same value must be configured for the mail relay.
<code>effektif.google.secrets.location</code>	(Optional) Absolute path to the Google APIs OAuth 2.0 credentials file, see Configure Google service actions .

The following properties are only relevant if your Workflow Accelerator installation is connected to your Signavio installation.

```
effektif.process.manager.serviceBaseUrl  
effektif.process.manager.authentication.clientId  
effektif.process.manager.authentication.clientSecret  
effektif.process.manager.authentication.jwt.secret
```

See [Signavio Process Manager integration](#) for more information on how to set up the integration.

2.12.3 Configure logging

The Workflow Accelerator system has its own application log file `effektif.log`. You need to define the directory where this log file is stored. The directory is specified in the file `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/logback.xml`. Open the file and update the following line:

```
<property name="LOG_DIR" value="./log"/>
```

The value should contain an absolute path. Make sure the user executing Tomcat has the required read / write permissions for the directory.

On Windows it necessary to escape all "\". An example value could like this:

```
"C:\\Logs\\Signavio".
```

This configuration file also enables you to change the log level in this file. By default, the log level is INFO. However, in a scenario where you require official support, you will be asked to set the log level to DEBUG. In order to so, update the following:

```
<root level="INFO">
```

Change it to:

```
<root level="DEBUG">
```

You will need to restart the Tomcat server for the changes to take effect. You can find more detailed information on how to start the Tomcat server in the next section.

The log file is configured to do a roll over. Every day, a new log file is created which contains a timestamp in the filename similar to `effektif-2014-10-09.log`. The most recent log entries can still be found in the file `effektif.log`. After 30 days, the log files will be removed automatically.

3 Optional configuration

This section provides instructions for optional configuration, which is not required for all installations.

3.1 User and group synchronisation with Active Directory

The enterprise version of Signavio Workflow Accelerator offers user and group synchronisation with an Active Directory Server using the LDAP V3 protocol. Before you can configure the synchronisation, check that you have the following pre-requisites.

- Workflow Accelerator is installed.
- The first Workflow Accelerator user is registered, see [Create the initial user](#) .
- A domain account for querying the Active Directory - its username and password.

Configuring the synchronisation consists of the following steps.

1. Create an Active Directory group that contains all Workflow Accelerator users as members.
2. Create an Active Directory group that contains all Workflow Accelerator administrators as members.
3. (Optional) Create an Active Directory group that contains all groups as members.
4. Log in to Workflow Accelerator with your initial user.
5. Open the *Organization settings* and select the *LDAP* tab.
6. Create a new Active Directory configuration.
7. Validate your configuration.
8. Synchronise the users and groups.

3.1.1 Creating the Active Directory groups

Active Directory users and groups that should be synchronised with Workflow Accelerator are managed within Active Directory by adding them to specific Active Directory groups. There are two groups for users, one which contains all Workflow Accelerator users and one which contains only administrators. All members of the administrator group must be members of the user group as well. Nested groups are not supported. All users and administrators need to be direct members of their groups.

The synchronisation will copy the attributes `givenName`, `sn`, `mail`, `dn`, `objectGUID`, `sAMAccountName` and `userPrincipalName` to the user in Workflow Accelerator. It is required that every user has the `mail` attribute defined.

Optionally, you can define a third group that contains all Active Directory groups that should be synchronised. All members of this group should be Active Directory groups as well. Again nested groups are not supported. Users must be direct members of the synchronised groups. The synchronisation will copy the attributes `cn`, `dn` and `objectGUID` to the group in Workflow Accelerator. Every group member that is not a Workflow Accelerator user will be ignored.

3.1.2 Creating a new configuration

In the LDAP configuration tab, select *Create* to start the configuration. Enter the configuration details on the configuration form.

- *URL* - the location of your Active Directory server, e.g. `ldap://adtest.local:389`
- *username* and *password* - the login credentials of the domain account
- *User group DN* - the distinguished name (DN) of the Active Directory group which contains all users
- *Admin group DN* - the DN of the Active Directory group which contains all administrators
- *Group group DN* - (optional) the DN of the Active Directory group which contains all groups
- *License* - the license that will be assigned to every synchronised user

The last field of the configuration form will show you if the configuration was already validated.

3.1.3 Validating a configuration

Before synchronisation can start, the configuration must be validated. Validation includes the following checks.

- Workflow Accelerator can connect to the Active Directory.
- The configured groups are proper Active Directory groups.
- There is at least one user.
- There is at least one administrator.
- The users have the `mail` attribute defined.
- There are sufficient licenses.
- (Optional) There are groups defined.

Start the validation by selecting *Validate*. If the configuration is valid, the *Synchronise* button is enabled. If the configuration is not valid, the page shows a list with more detailed information. In that case fix the issues and run the validation again.

Every time you change the configuration, the validation must be executed again. Note the validation might take several seconds depending on the number of users you want to synchronise and the performance of the Active Directory server.

3.1.4 Synchronising users and groups

If the configuration is valid, you can synchronise the users and groups. Start the synchronisation by clicking on *Synchronise*. If the synchronisation was successful, you will see a report of the added / updated users and groups. If the synchronisation failed, a list with more detailed information will be shown.

Note the synchronisation might take several seconds or minutes depending on the number of users you want to synchronise and the performance of the Active Directory.

3.2 Single Sign-On using Windows and Kerberos/SPNEGO

Use this configuration to enable Single Sign-On (SSO) in a Windows environment that supports user authentication using **Kerberos** and **SPNEGO**.

Before you start, check that you have the following pre-requisites.

- Windows domain users added to Workflow Accelerator, e.g. using LDAP synchronisation.
- A Windows domain for authenticating users.
- A domain account for running Workflow Accelerator - its username and password.
- Workflow Accelerator is installed on a Windows Server that is on the domain.

Configuring SSO in Apache Tomcat consists of the following steps.

1. Install the SPNEGO library's JAR file.
2. Configure the SPNEGO SSO Servlet filter.
3. Add the `krb5.conf` and `login.conf` configuration files to `$TOMCAT_HOME/`.
4. Register Tomcat's Service Principal Name (SPN)
5. Configure Tomcat to be executed with a domain account.
6. Restart Tomcat.

3.2.1 Install the SPNEGO library

The SPNEGO library makes it possible to authenticate users by connecting to an Active Directory server.

To install, copy the following file to `$TOMCAT_HOME/lib/`:

```
$WORKFLOW_HOME/optional/spnego/spnego.jar
```

3.2.2 Configuring the SPNEGO SSO Servlet filter

The SSO Servlet filter uses Windows authentication to automatically log in when a user accesses Workflow Accelerator.

To configure, edit `$TOMCAT_HOME/conf/web.xml` and add the following lines immediately before the final line (`</web-app>`), from `$WORKFLOW_HOME/optional/spnego/filter-configuration.xml`.

```
<filter>
  <filter-name>SpnegoHttpFilter</filter-name>
  <filter-class>net.sourceforge.spnego.SpnegoHttpFilter</filter-class>
  <init-param>
    <param-name>spnego.allow.basic</param-name>
    <param-value>>false</param-value>
  </init-param>
</init-param>
```



```

    <param-name>spnego.allow.localhost</param-name>
    <param-value>>false</param-value>
</init-param>
<init-param>
    <param-name>spnego.allow.unsecure.basic</param-name>
    <param-value>>false</param-value>
</init-param>
<init-param>
    <param-name>spnego.login.client.module</param-name>
    <param-value>spnego-client</param-value>
</init-param>
<init-param>
    <param-name>spnego.krb5.conf</param-name>
    <param-value>krb5.conf</param-value>
</init-param>
<init-param>
    <param-name>spnego.login.conf</param-name>
    <param-value>login.conf</param-value>
</init-param>
<init-param>
    <param-name>spnego.login.server.module</param-name>
    <param-value>spnego-server</param-value>
</init-param>
<init-param>
    <param-name>spnego.prompt.ntlm</param-name>
    <param-value>>false</param-value>
</init-param>
<init-param>
    <param-name>spnego.logger.level</param-name>
    <param-value>1</param-value>
</init-param>
<init-param>
    <param-name>spnego.preauth.username</param-name>
    <param-value>??</param-value>
</init-param>
<init-param>
    <param-name>spnego.preauth.password</param-name>
    <param-value>??</param-value>
</init-param>
</filter>

<filter-mapping>
    <filter-name>SpnegoHttpFilter</filter-name>
    <url-pattern>/api/v1/users/auth</url-pattern>
</filter-mapping>

```

Near the bottom, replace two occurrences of `???` with the domain account's user-name and password, respectively.

3.2.3 Adding SPNEGO configuration files

Copy `krb5.conf` to `$TOMCAT_HOME/`, and edit the file:

- replace five occurrences of `ADTEST` with the Windows domain name
- replace `123.45.6.7` with the host name or IP address of the Active Directory server.

3 Optional configuration

```
[libdefaults]
    default_tkt_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
    default_tgs_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
    permitted_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
    default_realm = ADTEST.LOCAL

[realms]
    ADTEST.LOCAL = {
        kdc = 123.45.6.7
        default_domain = ADTEST.LOCAL
    }

[domain_realm]
    .ADTEST.LOCAL = ADTEST.LOCAL
```

Copy `login.conf` to `$TOMCAT_HOME/`.

```
spnego-client {
    com.sun.security.auth.module.Krb5LoginModule required;
};

spnego-server {
    com.sun.security.auth.module.Krb5LoginModule required
    storeKey=true
    isInitiator=false;
};
```

3.2.4 Registering Tomcat's Service Principal Name (SPN)

The **SPN** is the identifier used by Kerberos authentication to associate a service instance with a service logon account. You will have to associate the domain account for Workflow Accelerator with the service's fully-qualified domain name (FQDN). Therefore, you should register a SPN for every possible alias of the service, including the computer name of the server with and without domain name, as well as any other DNS entries that point to this server. In order to define a SPN, use `setspn.exe` - part of the **Windows Support Tools**.

```
setspn.exe -A HTTP/computer-name.domain-name domain-user-account
```

For example, our domain is called `ADTEST.LOCAL`, the computer name of the server is `VM-42`, Workflow Accelerator will be available under the domain `http://workflow.intranet:8080` and the domain user account is `tomcat@adtest.local`. Then we would have to register the following SPNs:

```
setspn.exe -A HTTP/vm-42 tomcat
setspn.exe -A HTTP/vm-42.adtest.local tomcat
setspn.exe -A HTTP/workflow.intranet tomcat
```

For more information see the [Microsoft documentation](#).

You can list all registered SPNs which are associated to a domain account with the following command:

```
setspn.exe -L domain-user-account
```

3.2.5 Configuring the Tomcat user

By default the Tomcat application server is executed with a local system account. The SSO setup requires the Tomcat to be executed with a domain account. You should use the same domain account which you already configured in [Configuring the SPNEGO SSO Servlet filter](#).

1. Open the Windows services dialogue `services.msc`
2. Select your Apache Tomcat service and open the properties
3. Select the **Log On** tab
4. Choose the option *This account* and fill in the domain account credentials

3.2.6 Troubleshooting

First follow the steps in the [SPNEGO Pre-flight checklist](#). Run the `HelloKDC` connection test in the `$TOMCAT_HOME/` directory to use the `krb5.conf` and `login.conf` you configured above.

If running `HelloKDC` results in error message like "KDC has no support for encryption type (<number>)", make sure that the user account for the Active Directory user being signed in has these two options disabled

- *Use only Kerberos DES encryption types for this account*
- *This account supports Kerberos AES 256 bit encryption*

and the option *This account supports Kerberos AES 128 bit encryption* enabled. Note that in the *Pre-flight checklist*, the *Do we know the address of the KDC?* section recommends using the KDC host name instead of the IP address.

If you have completed the setup described here and the SSO is still not working, please make sure you are accessing the web application from a different machine than the one that runs the Tomcat server. Furthermore, check if **Integrated Windows Authentication** is activated and the Workflow Accelerator system is a trusted service, for instance by adding the address to the local intranet. If you see an error page with the following error message:

```
GSSException: Failure unspecified at GSS-API level (Mechanism level: Clock skew too great (37))
```

this means the time difference between the computer accessing Workflow Accelerator and the Kerberos server exceeds 5 minutes. You need to synchronise the time on all machines, e.g. by running `ntp` to fix the issue.

Additionally, you can check the log files in `$TOMCAT_HOME/logs` for further error messages. These error messages may offer hints about why the SSO set-up is failing.

If the log files contain the error message `Request header is too large`, the size of the Kerberos tickets issued by your KDC and sent in the request header are too large for the default Tomcat settings. You have to set the `maxHttpHeaderSize` option in the Tomcat `server.xml`.

1. Open `$TOMCAT_HOME/conf/server.xml`
2. Locate the `Connector` configuration in the file
3. Add the value `maxHttpHeaderSize="65536"`:

```
<Connector port="8080" maxHttpHeaderSize="65536" protocol="HTTP/1.1" ... />
```

4. Save the file and restart Tomcat

3.3 Configure Google service actions

To use Google actions within Workflow Accelerator you need to create your own OAuth 2.0 client credentials with Google to access the Google APIs.

Your Workflow Accelerator system must be available on the public internet and the URL must end with a public top level domain.

3.3.1 Create OAuth 2.0 client credentials for Google services

The following Google article explains how to create OAuth 2.0 client credentials:

https://support.google.com/googleapi/answer/6158849?hl=en&ref_topic=7013279

1. Setup a new project in <https://console.developers.google.com>
2. Create OAuth client credentials
 - When asked for the application type, choose *web application*
 - For *Authorised redirect URIs* use the base URL you configured in your Workflow Accelerator configuration file and add the following path `/api/v1/oauth_callback`, e.g. `https://workfow.example.com/api/v1/oauth_callback`. The redirect URI is required to successfully add new Google accounts to a user's Workflow Accelerator account.
3. Activate the APIs required by Workflow Accelerator (see the following section for more information)
4. Update the user consent screen
 - Whenever your users add their Google accounts to get access to the Google services in Workflow Accelerator they will see this consent screen

3.3.2 Activate required Google APIs

Once you created the Google APIs project and setup the client credentials, you need to activate the Google APIs for your project which are used by Workflow Accelerator.

1. Open <https://console.developers.google.com/apis/dashboard> - your project's dashboard and click *Activate APIs and services*
2. Search for the following three APIs and activate them
 - Google Drive API
 - Google Sheets API

3 Optional configuration

- Google Calendar API
- 3. Once you are finished you should see them listed on your project dashboard

3.3.3 Download and install the client credentials

Workflow Accelerator needs to be configured to use your Google client credentials.

1. Open <https://console.developers.google.com/apis/credentials> - your Google APIs project's credentials page
2. You should see your previously created OAuth 2.0 credentials for Workflow Accelerator listed there
3. Next to the entry click *Download JSON*
4. The downloaded file has a long name which includes a dynamic part. If you like you can rename it to something shorter, e.g.
`google_client_credentials.json`
 - For the rest of this guide we will assume the file is called `google_client_credentials.json`. Please adjust any following examples to your chosen filename.
5. Upload the JSON file to the Workflow Accelerator server
6. The file must be accessible by the Tomcat server as it will be loaded on startup of the application
 - You can use for instance the Tomcat `conf` directory
7. Add the absolute path of the JSON file to the Workflow Accelerator configuration file
 - In `effektif-onpremise.properties` add the entry
`effektif.google.secrets.location = /path/to/google_client_credentials.json`
 - Replace the value with the absolute path to your JSON file
8. Restart Tomcat

When you open the workflow editor, you should now be able to add Google actions to your workflows and your Google accounts in the actions configuration.

3.3.4 Apply for application verification

The Google integration accesses sensitive data like files in Google Drive and calendars. Google introduced a verification mechanism for applications that use OAuth to access sensitive data. Unverified apps will have stricter quota limits for API calls and users will be warned when granting access to an unverified application.

The following article gives more information about applications that need verification and the verification process itself:

<https://support.google.com/cloud/answer/7454865?hl=en>

Please read the article and check whether your Google APIs project (application) needs to be verified.

3.4 Implement a configuration provider

Workflow Accelerator can be configured by using either a configuration file (see section [Configuring Workflow Accelerator](#)) or system properties. However, those configurations are static and will exist in plain text on your server. As a third option you can implement your own configuration provider and add it to Workflow Accelerator. The configuration provider can specify a value for all available configuration options ([All configuration options](#)). The configuration provider only needs to provide those configuration values that you don't want to configure in your configuration file.

The configuration provider library below is for Java. Your implementation must be compatible with Java 8 or 11.

3.4.1 Create your own configuration provider

Implement the interface `com.signavio.workflow.config.ConfigurationProvider` from `com.signavio.workflow:config-provider` library. You find the interface [here](#).

1. Add the dependency to your project. Example for Apache Maven:

```
<dependency>
<groupId>com.signavio.workflow</groupId>
<artifactId>config-provider</artifactId>
```

3 Optional configuration

```
<version>1.0.2</version>
</dependency>
```

2. Create a class that implements `ConfigurationProvider` and overrides the method `Map<String, String> get()`:

src/main/java/com/signavio/workflow/config/ExampleConfigProvider.java

```
import java.util.*;
import com.signavio.workflow.config.ConfigurationProvider;

public class ExampleConfigProvider implements ConfigurationProvider {

    @Override
    public Map<String, String> get() {
        return Collections.unmodifiableMap(
            new HashMap<String, String>() {{
                put("effektif.mongodb.password", "test_password");
                put("key", "value");
            }});
    }
}
```

3. Save the generated implementation as a JAR file.

3.4.2 Configure Workflow Accelerator to use your configuration provider

1. Add a file to `META-INF/services`. The file name should be the fully qualified service name without any extension `com.signavio.workflow.config.ConfigurationProvider`.
2. In the file, specify the fully qualified implementation:
`src/main/resources/META-INF/services/com.signavio.workflow.config.ConfigurationProvider`
`com.signavio.workflow.config.ExampleConfigProvider`
3. Copy the JAR file you created earlier to the directory `$TOMCAT_HOME/webapps/ROOT/WEB-INF/lib`.
4. Restart Workflow Accelerator to apply your changes.

If the service configuration was loaded successfully, you see an entry in the catalina log:

```
13:08:24.144 INFO com.effektif.CustomServiceConfiguration Loading configuration from com.signavio.config.ExampleConfigProvider
```

If multiple service configurations are found, you see the following error message:

```
Found more than 1 configuration provider: com.signavio.workflow.config.ExampleConfigProvider, com.signavio.workflow.config.OtherConfigProvider
```


4 Signavio Process Manager integration

The Signavio Process Manager integration with Signavio Workflow Accelerator enables Process Manager users to execute approval workflows for their process models using Workflow Accelerator. In order to setup the integration properly, both systems, Process Manager and Workflow Accelerator, have to be installed. For information about installing Process Manager refer to Process Manager's administration manual. In case both systems run on the same server, you need to set up two Tomcat instances.

4.1 Compatibility list

To ensure optimum performance, you need to use a compatible version of Process Manager in combination with your Workflow Accelerator system. The following table shows the compatible versions of Workflow Accelerator and Process Manager and the compatible versions for required additional software.

Version numbers

Workflow Accelerator	Process Manager	Java	Node.js	MongoDB	Tomcat	Upgrade from Workflow Accelerator	Notes
3.146	14.9	8-11 we recommend to use AdoptOpenJDK 8 (LTS)	12	3.4 - 4.2	8.5	3.72	
3.129	13.15	8-11 we recommend to use AdoptOpenJDK 8 (LTS)	12	3.4 - 4.2	8.5	3.72	
3.99	13.3	8	8 - 10	3.4 - 4.0	8.5	3.72	
3.85	12.12	8	8 - 10	3.4 - 3.6	8.5	3.22	
3.72	12.3.3	8	8	3.4 - 3.6	8.5	3.22	
3.56		8	6 - 8	3.4 - 3.6	8.0 - 8.5	3.22	

4.2 Setup

1. Make sure both systems are NOT running.
2. Edit the Process Manager configuration file, which can be found in the Tomcat that hosts the Process Manager system at the following path:
 - version 9.3 and earlier: `WEBAPPS/ROOT/WEB-INF/classes/configuration.xml`
 - version 9.7 and later: `CONF/configuration.xml`

```
<effektif>
  <effektifServerUrl>http://your.workflow.domain:port</effektifServerUrl>
  <effektifUserName>system@signavio.com</effektifUserName>
  <effektifUserPw>yourpassword</effektifUserPw>
</effektif>

<sharedJwtSecret>SHARED_SECRET_CHANGE_ME</sharedJwtSecret>>
```

<code>effektifServerUrl</code>	Defines the URL under which your Workflow Accelerator system is available.
<code>effektifUserName</code>	Defines the mail of the system user that will access the Workflow Accelerator system. Make sure to use <code>system@signavio.com</code> to ensure the integration to work properly. The value is already preconfigured. You do NOT have to create an account for this user in any of the systems.
<code>effektifUserPw</code>	Defines the password of the system user. Remember this password: you will need to use the exact same password in the Workflow Accelerator configuration file as <code>effektif.process.manager.authentication.clientSecret</code>
<code>sharedJwtSecret</code>	Defines a shared secret that is used to secure the communication between Workflow Accelerator and Process Manager. You will need to set the exact same secret in the Workflow Accelerator configuration file as <code>effektif.process.manager.authentication.jwt.secret</code> .

3. Edit the Workflow Accelerator configuration file `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/effektif-onpremise.properties` and add / update the following entries:

```
effektif.process.manager.BaseUrl=http://your.signavio.domain:port
```

4 Signavio Process Manager integration

```
effektif.process.manager.  
authentication.clientId=com.effektif.oem  
  
effektif.process.manager.  
authentication.clientSecret=yourpassword  
  
effektif.process.manager.  
authentication.jwt.secret=SHARED_SECRET_CHANGE_ME
```

<code>effektif.process.manager.baseUrl</code>	Defines the URL under which your Process Manager system is available.
<code>effektif.process.manager.authentication.clientId</code>	Defines the ID the Workflow Accelerator system uses to identify itself to the Process Manager system. Make sure to use <code>com.effektif.oem</code> to ensure the integration to work properly. This value is already preconfigured.
<code>effektif.process.manager.authentication.clientSecret</code>	Defines the password the Workflow Accelerator system uses to identify itself to the Process Manager system. Reuse the same password that you have set in the Process Manager configuration file as <code>effektifUserPw</code>
<code>effektif.process.manager.authentication.jwt.secret</code>	Defines a shared secret that is used to secure the communication between Workflow Accelerator and Process Manager. Reuse the same secret that you have set in the Process Manager configuration file as <code>sharedJwtSecret</code>

4. Start both systems.
5. Activate the approval workflow feature within your Process Manager workspace. Your Workflow Accelerator organization will be setup automatically. For more information on how to activate the approval workflow, refer to Process Manager's user manual.

The configuration parameters have been renamed with version 3.56. See the following table for the new names. The old names are now deprecated, but will still work until further notice.

deprecated name	new name
<code>effektif.com.effektif.model. service.signavio. SignavioAuthenticationProvider. serviceBaseUrl</code>	<code>effektif.process. manager.baseUrl</code>
<code>effektif.com.effektif.model. service.signavio. SignavioAuthenticationProvider. clientId</code>	<code>effektif.process.manager. authentication.clientId</code>
<code>effektif.com.effektif.model. service.signavio. SignavioAuthenticationProvider. clientSecret</code>	<code>effektif.process.manager. authentication.clientSecret</code>

4.3 User management

If you have purchased the OEM version of Workflow Accelerator that is shipped with Process Manager, user management will be handled by the Process Manager system. The Process Manager system will synchronize all users to Workflow Accelerator. Within Workflow Accelerator you will not be able to register any initial user nor invite more users.

If you have purchased a standalone version of Workflow Accelerator, you will be able to create an initial user and invite more users to Workflow Accelerator. However, we advise that you create your Workflow Accelerator organization by activating the approval workflow feature in Process Manager. This way, your Process Manager workspace and Workflow Accelerator organization will be linked automatically.

5 Updating Signavio Workflow Accelerator

This section will guide you through an update of your Workflow Accelerator system. It assumes you have successfully installed an early version of Workflow Accelerator. If you are installing Workflow Accelerator for the first time, you can skip this chapter.

First, check **Update notes for specific versions** for additional information about the update to the latest version of Workflow Accelerator before continuing.

5.1 Unpack Signavio Workflow Accelerator

Unzip `signavio-workflow-{version}.zip` to a directory of your choice, e.g. `/yourlocation/signavio-workflow-{version}`. In the rest of this guide, we'll refer to the Workflow Accelerator home directory as `$WORKFLOW_HOME`. The directory will contain the following files:

- *.sh - Linux shell scripts
- db - directory that contains MongoDB configuration files
 - mongod.conf - configuration file for Linux
 - mongod.cfg - configuration file for Windows
- mail-relay - directory that contains the mail relay
- script-engine - directory that contains the script engine
- optional - directory that contains optional resources
 - spnego - directory that contains resources to setup SSO with SPNEGO
- webapps - directory
- ROOT - this directory contains the web application

On Linux systems you need to make the scripts executable with the `chmod` command like this:

```
chmod a+x $WORKFLOW_HOME/*.sh
```

5.2 Stop Workflow Accelerator

Before you can start with the update process you have to shutdown the Workflow Accelerator system. It is sufficient when you shutdown the Tomcat server Workflow Accelerator is running on. You don't have to shutdown the MongoDB server. If you installed the Workflow Accelerator script engine, you need to stop it as well. Refer to chapter [Stopping Signavio Workflow Accelerator](#) for information on how to stop Workflow Accelerator.

5.3 Backup MongoDB

Updating Workflow Accelerator to a newer version might involve changes in the database schema. Therefore, you are requested to backup the database before you go on. See [Backup](#) for more information on how to create a backup. Furthermore, the backup is necessary in case you have to revert Workflow Accelerator to an older version, because changes of the database schema are in general not backward compatible. This means you cannot run an older version of Workflow Accelerator with a database that was already migrated to a newer version.

5.4 Backup the configuration files

During the first setup you have modified some configuration files and added the Workflow Accelerator license file. In order to preserve those changes, you have to copy those files to another location before updating the Workflow Accelerator system. You can find those files in the directory `$TOMCAT_`

`HOME/webapps/ROOT/WEB-INF/classes`.

<code>effektiv-onpremise.properties</code>	Workflow Accelerator configuration file
<code>logback.xml</code>	configuration file for Workflow Accelerator logs
<code>license.xml</code>	Workflow Accelerator license file

Simply copy the files mentioned above to another location, e.g. `$TOMCAT_`

`HOME/webapps`.

If you installed the Workflow Accelerator script engine, you modified the configuration file `configuration.onpremise.js` of the script engine as well. You can find the configuration file directly within the `SCRIPT_ENGINE_HOME` directory. Please copy it as well to another location before you go on.

5.5 Install a new version

In order to update the Workflow Accelerator system, you have to replace the `ROOT` folder in `TOMCAT_HOME/webapps` with the newer version you can find in the extracted directory `WORKFLOW_HOME/webapps`.

1. Delete the directory `TOMCAT_HOME/webapps/ROOT` completely.
2. Copy the directory `WORKFLOW_HOME/webapps/ROOT` to `TOMCAT_HOME/webapps`.
3. Copy the configuration files you have backed up in subsection **Backup the configuration files** back into the directory `TOMCAT_HOME/webapps/ROOT/WEB-INF/classes`.
4. If you have been provided with a new `license.xml`, copy the file into `TOMCAT_HOME/webapps/ROOT/WEB-INF/classes` and replace the existing file. Otherwise, you can keep on using the old `license.xml`.

If you have installed the Workflow Accelerator script engine, you need to update the content of the `SCRIPT_ENGINE_HOME` directory with the newer version you can find in the extracted directory `WORKFLOW_HOME/script-engine`.

1. Delete all files in the directory `SCRIPT_ENGINE_HOME`.
2. Copy the content of the directory `WORKFLOW_HOME/script-engine` to your `SCRIPT_ENGINE_HOME` directory.
3. Copy the configuration file `configuration.onpremise.js` you have backed up in subsection **Backup the configuration files** back into the directory `SCRIPT_ENGINE_HOME`.

If you have installed the Workflow Accelerator mail relay, you need to update the content of the `mail-relay.jar` file within the `MAIL_RELAY_HOME` directory with the newer version you can find in the extracted directory `WORKFLOW_HOME/mail-relay`.

1. Delete the file `$MAIL_RELAY_HOME/mail-relay.jar`
2. Copy the file `$WORKFLOW_HOME/mail-relay/mail-relay.jar` to your `$MAIL_RELAY_HOME` directory.

5.6 Revisit the configuration files

After upgrading to a newer version, it might be necessary to update existing configuration files and add, remove, or modify some values. Refer to section **Configuring Workflow Accelerator** and make sure your Workflow Accelerator configuration file is up-to-date before starting the system again.

5.6.1 Reconfigure your optional configuration provider

You can implement your own configuration provider and add it to Workflow Accelerator, see section **Implement a configuration provider**. If you use a configuration provider, you need to reconfigure it every time you update Workflow Accelerator.

5.7 Start Workflow Accelerator

When you have successfully installed the new Workflow Accelerator version and verified that the configuration files are up-to-date, you can restart the Workflow Accelerator system. Refer to chapter **Starting Signavio Workflow Accelerator** for more information on how to start Workflow Accelerator.

During the first startup the Workflow Accelerator system will migrate the database if there are any outstanding schema changes. Therefore, the startup might take a short while depending on the size of your database. If the system doesn't become available again within minutes, please check the Workflow Accelerator log file for any new error reports.

5.8 Restore an older version

In case of exceptional problems you might want to restore an older version of Workflow Accelerator. Be aware, if you revert Workflow Accelerator to an older version, it is not guaranteed that you can keep on using your current database. The upgrade to a newer version can involve schema changes that render the database incompatible with older versions of Workflow Accelerator. In this situation, the Workflow Accelerator system will not start if it detects a newer version of the database schema which it doesn't support. This is the reason why you should create a backup before updating the system.

In order to restore an older version, you can follow the instructions in this chapter again. However, instead of creating a backup you should restore an older backup.

1. Unpack the older version of Workflow Accelerator, see [Unpack Signavio Workflow Accelerator](#)
2. Stop the Workflow Accelerator system, see [Stopping Signavio Workflow Accelerator](#)
3. Restore an earlier back up that was taken while the old Workflow Accelerator version was deployed. See [Restoring backups](#) for more information on how to restore a backup.
4. Backup the configuration files, see [Backup the configuration files](#)
5. Install the older version of the webapp in `$TOMCAT_HOME/webapps`, see [Install a new version](#)
6. Install the older version of the script engine in `$$SCRIPT_ENGINE_HOME`, see [Install a new version](#)
7. Install the older version of the mail relay in `$MAIL_RELAY_HOME`, see [Install a new version](#)
8. Check the configuration files again, see [Revisit the configuration files](#)
9. Start the Workflow Accelerator system, see [Starting Signavio Workflow Accelerator](#)

Now, the Workflow Accelerator system should be running with the older version again.

6 Update notes for specific versions

This section contains important information about Signavio Workflow Accelerator system updates. It lists changes to technical requirements for new versions and additional instructions for updates.

If you skip one or more versions of Workflow Accelerator during the update, please follow the update notes for the versions you are skipping as well. You should start with the oldest version which is newer than your currently installed version of Workflow Accelerator.

6.1 Updating to Signavio Workflow Accelerator 3.146

To update Workflow Accelerator from a version prior to 3.72, you must first update the system to version 3.72 before you can update to version 3.146. **Download Signavio Workflow 3.72** and follow the update notes (**Updating to Signavio Workflow Accelerator 3.72**) before you continue.

If you skip one or more versions of Workflow Accelerator for the update, also follow the update notes for all versions you are skipping.

6.1.1 For Signavio Process Manager integration

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 14.9.

6.2 Updating to Signavio Workflow Accelerator 3.129

To update Workflow Accelerator from a version prior to 3.72, you must first update the system to version 3.72 before you can update to version 3.129. **Download Signavio Workflow 3.72** and follow the update notes (**Updating to Signavio Workflow Accelerator 3.72**) before you

continue.

If you skip one or more versions of Workflow Accelerator for the update, also follow the update notes for all versions you are skipping.

6.2.1 For Signavio Process Manager integration

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 13.15.

6.2.2 Updating Node.js

For this version of Workflow Accelerator, update to Node.js 12.x LTS

1. **Download the required Node.js LTS version**
2. Stop your Workflow Accelerator script engine - see section **Stopping Signavio Workflow Accelerator** .
3. Uninstall your old Node.js version
4. Install the new Node.js LTS version
5. (Optional) Update your startup script / service for Workflow Accelerator script engine if the install directory for Node.js changed.
 - *Windows* : If you are using NSSM to run the service, open the command line `cmd` with administrative privileges, execute `nssm edit script-engine` and update the location of Node.js in the dialog.
6. Start Workflow Accelerator script engine - see section **Starting Signavio Workflow Accelerator** .

6.2.3 New Mongo DB version supported

Workflow Accelerator 3.129 now supports MongoDB version 4.2.x which comes with improved performance and functionality.

Workflow Accelerator does not support MongoDB version 4.4.x.

Workflow Accelerator still supports MongoDB versions 3.4.x and 3.6.x . Therefore, an upgrade to MongoDB 4.2.x is not yet required.

6.2.4 Updating the configuration for Single Sign-On using Windows and Kerberos/SPNEGO

If you have configured SSO via Kerberos/SPNEGO for Workflow Accelerator, you need to update the configuration.

1. Ensure Tomcat is stopped.
2. Open `$TOMCAT_HOME/lib`.
3. Remove the file `signavio-cfc.jar`
4. Open `$TOMCAT_HOME/conf/web.xml` in a text editor.
5. Locate and remove the following filter configuration:

```
<filter>
  <filter-name>cfc</filter-name>
  <filter-class>ConditionalFilterChain</filter-class>
</filter>
<filter-mapping>
  <filter-name>cfc</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

6. Locate the following filter mapping:

```
<filter-mapping>
  <filter-name>SpnegoHttpFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

7. Change the value of `url-mapping` to `/api/v1/users/auth`:

```
<filter-mapping>
  <filter-name>SpnegoHttpFilter</filter-name>
  <url-pattern>/api/v1/users/auth</url-pattern>
</filter-mapping>
```

You don't have to change configuration of the `SpnegoHttpFilter` .

6.3 Updating to Signavio Workflow Accelerator 3.99

To update Workflow Accelerator from a version prior to 3.72, you must first update the system to version 3.72 before you can update to version 3.99. **Download Signavio Workflow 3.72** and follow the update notes (**Updating to Signavio Workflow Accelerator 3.72**) before you continue.

If you skip one or more versions of Workflow Accelerator for the update, also follow the update notes for all versions you are skipping.

6.3.1 For Signavio Process Manager integration

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 13.3.

6.3.2 New Mongo DB version supported

Workflow Accelerator 3.99 now supports MongoDB version 4.0.x which comes with improved performance and functionality.

Workflow Accelerator does not support MongoDB version 4.2.x.

Workflow Accelerator still supports MongoDB versions 3.4.x and 3.6.x . Therefore, an upgrade to MongoDB 4.0.x is not yet required.

6.4 Updating to Signavio Workflow Accelerator 3.85

6.4.1 For Signavio Process Manager integration

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 12.12.

From this version on, it is necessary to set the `sharedJwtSecret` in both the Workflow Accelerator configuration file and the Process Manager configuration file. See **Setup** for details.

6.4.2 Data format of data sent from connectors

Version 3.85 introduces a validation for data retrieved via a custom data connector.

If you are using a custom data connector which stores **records by value**, verify that the data type and the data format provided by the connector matches the data specified by your connector descriptor (Read more about **data types and**

formats). Data that does not match is no longer accepted. If the connector sends mismatching data, users are not able to select the relevant records from the drop-down list but will see an error message.

6.4.3 Updating Node.js

For this version of Workflow Accelerator, update to Node.js 10.x LTS

1. **Download the required Node.js LTS version**
2. Stop your Workflow Accelerator script engine - see section **Stopping Signavio Workflow Accelerator** .
3. Uninstall your old Node.js version
4. Install the new Node.js LTS version
5. (Optional) Update your startup script / service for Workflow Accelerator script engine if the install directory for Node.js changed.
 - *Windows* : If you are using NSSM to run the service, open the command line `cmd` with administrative privileges, execute `nssm edit script-engine` and update the location of Node.js in the dialog.
6. Start Workflow Accelerator script engine - see section **Starting Signavio Workflow Accelerator** .

6.5 Updating to Signavio Workflow Accelerator 3.72

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 12.3.

The Apache Tomcat Team announced that the support for Apache Tomcat 8.0.x has ended June 30, 2018. That's why Workflow Accelerator now only supports Apache Tomcat 8.5.x. If you are upgrading from Tomcat 8.0.x or older to Tomcat 8.5.x and configured SSL / TLS previously, then you will have to adjust your settings. The official Tomcat 8.5.x documentation explains how to configure SSL / TLS:

<https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html>

The Process Manager admin guide also contains instructions on how to setup SSL with Tomcat:

<https://docs.signavio.com/adminguide/editor/en/installation/https.html>

6.6 Updating to Signavio Workflow Accelerator 3.56

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 11.14. Furthermore, you must update the Workflow Accelerator configuration file to support new integration scenarios. See the following section for more information.

Workflow Accelerator 3.56 now supports MongoDB version 3.6.x which comes with improved performance and functionality. Workflow Accelerator still supports MongoDB 3.4.x. Therefore, an upgrade to MongoDB 3.6.x is not yet required.

If you use Workflow Accelerator script engine to execute custom JavaScript tasks in your workflows you are required to update Node.js to version 8 LTS (Long Term Support). In November 2017, Node.js 8 replaced Node.js 6 as the Long Term Support version. Workflow Accelerator 3.56 still supports Node.js 6. However, future versions will assume you are using Node.js 8. See the following section for more information.

6.6.1 Updating the configuration file

You only need to update the configuration file if your Workflow Accelerator system is integrated with Signavio Process Manager. To update the configuration open `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/effektif-onpremise.properties`. You can either update your existing configuration file or use the new version provided with the release and fill in your settings.

The following three entries have been renamed. You only need to change the part left of the `=` sign.

old name	new name
effektif.com. effektif.model. service.signavio. SignavioAuthenticationProvider. .serviceBaseUrl	effektif.process. manager.baseUrl
effektif.com. effektif.model. service.signavio. SignavioAuthenticationProvider. clientId	effektif. process.manager. authentication. clientId
effektif.com. effektif.model. service.signavio. SignavioAuthenticationProvider. clientSecret	effektif. process.manager. authentication. clientSecret

Furthermore, add the following new entry:

```
effektif.process.manager.authentication.jwt.secret = SHARED_SECRET_CHANGE_ME
```

The new value represents a secret that is used for the communication with Process Manager. Its value has to match the value you configured in Process Managers `configuration.xml` for the entry `sharedJwtSecret`.

6.6.2 Updating Node.js

For this version of Workflow Accelerator, update to Node.js 8.x LTS

1. **Download the required Node.js LTS version**
2. Stop your Workflow Accelerator script engine - see section **Stopping Signavio Workflow Accelerator** .
3. Uninstall your old Node.js version
4. Install the new Node.js LTS version
5. (Optional) Update your startup script / service for Workflow Accelerator script engine if the install directory for Node.js changed.
 - *Windows* : If you are using NSSM to run the service, open the command line `cmd` with administrative privileges, execute `nssm edit script-engine` and update the location of Node.js in the dialog.

6. Start Workflow Accelerator script engine - see section **Starting Signavio Workflow Accelerator** .

6.7 Updating to Signavio Workflow Accelerator 3.39

If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 11.5.

To update from a version prior to Signavio Workflow 3.22, first follow the instructions in the next sections version by version, starting with the oldest version that is higher than your currently installed version.

6.8 Updating to Signavio Workflow Accelerator 3.22

Signavio Workflow has been renamed to Signavio Workflow Accelerator. If you use Workflow Accelerator in combination with Signavio Process Manager, you must update Process Manager to version 10.15.

To update Workflow Accelerator from a version prior to 3.11, you **MUST** first update the system to version 3.15 before you can update to version 3.22. Download **Signavio Workflow 3.15** and follow the respective previous update instructions, before you continue with this section.

As well as upgrading the application, you should upgrade MongoDB. See the following section for more information.

6.8.1 Updating MongoDB

Workflow Accelerator 3.22 now supports the latest version of MongoDB (3.4.x) which comes with improved performance and functionality. You should upgrade to MongoDB version 3.4, as Workflow Accelerator releases will require it.

You can only update to MongoDB 3.4 if you are currently using version 3.2. The official MongoDB documentation contains for more information about the update:

<https://docs.mongodb.com/manual/release-notes/3.4-upgrade-standalone/>

To update your standalone MongoDB 3.2 to version 3.4 follow these steps:

1. Stop the Tomcat application server - see section **Stopping Signavio Workflow Accelerator** .
2. Create a backup of your database using `mongodump` - see section **Backup** .
3. Stop MongoDB - see **Stopping Signavio Workflow Accelerator** .
4. *Windows only*: manually remove the created MongoDB service by executing

```
$MONGO_HOME\bin\mongod.exe --remove
```

with administrator privileges on the command line `cmd`

5. Uninstall MongoDB 3.2.x, but ensure you don't delete the data directory and configuration file of your database as well as the backup!
6. Install MongoDB 3.4.x.
 - You can find a suitable version for your operating system in the MongoDB download center: <https://www.mongodb.com/download-center>
7. *Windows only*: Open the command line `cmd` with administrative privileges and execute the following command. Make sure to use absolute paths and replace them with the matching paths on your system.
 - `C:\MongoDB\bin\mongod.exe --config C:\MongoDB\mongod.cfg --install`
 - The referenced configuration file is the one you have created previously for the old version of MongoDB and should still be in the same location.
8. Start the new MongoDB instance.
9. Enable new functionality for version 3.4 by executing the following command on the command line.

6.8.1.1 Windows

1. Open a command line with administrative privileges
2. Navigate to the MongoDB bin directory
 - For example `C:\MongoDB\bin`
3. Execute the following command:
 - Replace `<username>` with your MongoDB user, e.g. `signavio`
 - Replace `<password>` with your MongoDB user password

```
mongo.exe admin -u <username> -p <password> --eval "db.adminCommand( { setFeatureCompatibilityVersion: '3.4' } )"
```

6.8.1.2 Debian

1. Open a command line
2. Execute the following command:
 - Replace `<username>` with your MongoDB user, e.g. `signavio`
 - Replace `<password>` with your MongoDB user password

```
mongo admin -u <username> -p <password> --eval "db.adminCommand({
setFeatureCompatibilityVersion: '3.4' })"
```

You finished the MongoDB update successfully.

6.9 Updating to Signavio Workflow 3.15

If you use Signavio Workflow in combination with the Signavio Process Editor, you must update the Signavio Process Editor to version 10.8.

To update from a version prior to Signavio Workflow 3.11, first follow the instructions in the next sections version by version, starting with the oldest version which is higher than your currently installed version.

6.10 Updating to Signavio Workflow 3.11

Effektif has been renamed to Signavio Workflow. Along with this change, the colors of the default theme have changed as well. You may notice the name of the database and database user in the examples in this manual have been changed from `effektif` to `signavio`. You don't need to change any of the names in your production environment. But keep in mind that you might have to adjust any examples in this manual to your actual naming.

With the new version of Signavio Workflow you are now able to configure a Node.js server to execute script tasks in a workflow. Note, you only need to configure this component if your version of Signavio Workflow supports script tasks. In general, you can use script tasks only with a full version of Signavio Workflow and not as part of the Signavio Approval Workflow. Refer to section **Installing the Workflow Accelerator script engine** for more information on how to setup the Signavio Workflow Script Engine.

If you update from a version prior to Effektiv 3.4, please follow the instructions in the next section to update your Java, Tomcat and MongoDB version as well. You don't need to do an intermediate update to Effektiv version 3.4 and can directly update to version 3.11 of Signavio Workflow.

6.11 Updating to Effektiv 3.4

Effektiv 3.4 requires several infrastructure changes. Starting with version 3.4, Effektiv requires Java 8. We also recommended upgrading MongoDB to version 3.2.x, because of the increased performance and functionality. Effektiv 3.4 will still support MongoDB 2.4.x, but later releases will assume that you are using MongoDB 3.2. Last but not least, Effektiv 3.4 supports Apache Tomcat 7 and 8. For now, you can use either version, but we recommend the optional upgrade to Tomcat 8.

If you are using Effektiv in combination with a Signavio system, you are required to update your Signavio system as well to version 9.7 or higher.

When updating Effektiv and your infrastructure, follow the following sequence:

1. Update Java to Java 8 (required)
2. Update Tomcat to Tomcat 8 (optional)
3. Update the Effektiv application
4. Update MongoDB to MongoDB 3.2 (required)

Be aware, the update to Effektiv 3.4 contains multiple changes in the underlying data schema which require several long running migrations. Depending on the size of your database, the migrations might take several seconds, or up to 20 minutes.

6.11.1 Updating Java and Tomcat

The last Effektiv release 2.27.x is already compatible with Java 8 and Tomcat 8. Start by updating Java and Tomcat, because Effektiv 3.4 will not run on Java 7. You can install Tomcat 8 next to an existing Tomcat 7, and then switch between the application servers. **Installing Tomcat** explains how to setup a Tomcat 8 server. After the successful setup, ensure none of the Tomcat servers is running.

Then copy the `$TOMCAT_HOME/webapps/ROOT` directory from your Tomcat 7 to your Tomcat 8 directory and start the Tomcat 8 server. Now, you should be able to access the Effektif system again via a browser.

6.11.2 Updating Effektif

As mentioned before, the update to version 3.4 changes the data schema.

Create a backup of your database before updating the application.

Data migrations might take from several seconds up to several minutes. These migrations are executed when the server starts, so the first start of Effektif might take longer than usual.

See [Updating Signavio Workflow Accelerator](#) for more information on how to update the Effektif application. After you have successfully updated Effektif, you can continue with the next section and update MongoDB.

6.11.3 Updating MongoDB

Effektif 3.4 now supports the latest version of MongoDB (3.2.x) which comes with improved performance and functionality. The usual upgrade from MongoDB 2.4.x to 3.2.x (<https://docs.mongodb.org/manual/release-notes/3.2-upgrade/>) requires a stepwise approach, first upgrading from 2.4 to 2.6, then from 2.6 to 3.0 and at last from 3.0 to 3.2. Upgrading the database also changes the storage engine from the old MMAPv1 engine to the new WiredTiger engine.

However, we will use a different approach to migrate your MongoDB 2.4.x to version 3.2.x. This approach will simply import a backup of MongoDB 2.4.x into a completely fresh setup of MongoDB 3.2.x.

1. Stop the Tomcat application server - see [Stopping Signavio Workflow Accelerator](#) .
2. Create a backup of your database using `mongodump` - see [Backup](#)
 - You will later import this backup into a new MongoDB 3.2 installation.
3. Stop MongoDB - see [Stopping Signavio Workflow Accelerator](#) .
4. *Windows only*: manually remove the created MongoDB service by executing

```
$MONGO_HOME\bin\mongod.exe --remove
```

with administrator privileges on the command line `cmd`

5. Uninstall MongoDB 2.4.x, but ensure you don't delete your backup!
 - *Windows:* Delete the MongoDB directory.
 - *Debian:* Use your package manager and uninstall the package `mongodb-10gen`.
6. Install MongoDB 3.2.x and recreate the Effektif database user again.
 - `$TOMCAT_HOME/webapps/ROOT/WEB-INF/classes/effektif-onpremise.properties` contains the current name and password of the Effektif database user
 - You can install the new version in the same place as the old one, please ensure the data directory of the new database is empty before you start it.
 - Note that the the MongoDB configuration file format has changed.
 - `install-mongodb` contains more information on how to setup MongoDB 3.2.x
7. Start the new MongoDB instance.
8. Restore the backup partially to your new MongoDB instance using `mongorestore`. Your backup from step 1 should contain multiple folders, one for each database in the MongoDB instance. You only need to restore the two that contain Effektif's data. Their default names are `effektif` and `effektif-files`. You might have chosen a different name by defining another value for `effektif.mongodb.database` in your Effektif configuration file. The value there represents the name of the first database and the name of the second database is generated by adding `-files`. You have to execute the restore command for both of the directories separately. The following lines show examples for the restore commands, assuming your MongoDB user and database are both called `effektif`.
 - The database name is specified by the `--db` parameter and also by the existing backup directory name.
 - *Windows:*
 - ```
$MONGO_HOME\bin\mongorestore.exe -u effektiv -p
<password> --authenticationDatabase admin --db effektiv
C:\path\to\dumps\effektif
```
    - ```
$MONGO_HOME\bin\mongorestore.exe -u effektiv -p
<password> --authenticationDatabase admin --db effektiv-files
C:\path\to\dumps\effektif-files
```

6 Update notes for specific versions

- *Debian:*

- ```
mongorestore -u effektiv -p <password>
--authenticationDatabase admin --db effektiv
/path/to/dumps/effektif
```

- ```
mongorestore -u effektiv -p <password>  
--authenticationDatabase admin --db effektiv-files  
/path/to/dumps/effektif-files
```

9. Restart Effektiv and MongoDB - see .

Your Effektiv system has been updated successfully. You can now access the Effektiv application again via a browser.

Note: if you use Robomongo to access MongoDB, download the latest version in order to be able to connect to MongoDB 3.2.

7 Starting Signavio Workflow Accelerator

If you have finished the setup successfully, you can go on and start the Workflow Accelerator application. You can use the checklist in the next subsection to verify you didn't forget anything. The following subsection explains how to start Workflow Accelerator on different operating systems.

7.1 Checklist

Before starting Workflow Accelerator, use the following checklist to make sure everything is setup properly.

- Java 8 or 11 is installed
- Apache Tomcat 8.5 is installed
 - the Java options are configured
- MongoDB is installed
 - MongoDB is configured
 - a MongoDB user for Workflow Accelerator was created
- the Workflow Accelerator ROOT folder was copied into the Tomcat webapps directory
- the Workflow Accelerator license was copied to the correct directory
- the Workflow Accelerator configuration file was updated
 - the base URL is set
 - the mail configuration is set
 - the database connection is set
- the Workflow Accelerator log file is configured

If your version includes the execution of JavaScript tasks:

- Node.js is installed
- the Workflow Accelerator script engine is installed
- the Workflow Accelerator script engine is configured
 - the port is set
 - the Workflow Accelerator URL is set
 - the log files are configured
 - the script engine URL is configured in the Workflow Accelerator configuration file

If your version includes the email trigger for workflows:

- the Workflow Accelerator mail relay is installed
- the Workflow Accelerator mail relay is configured
 - the port is set
 - the domain is set
 - the Workflow Accelerator URL is set
 - the log file is configured

If the system is integrated with Signavio Process Manager check also the following points:

- Workflow Accelerator and Signavio Process Manager have compatible versions
- Workflow Accelerator is configured in Signavio Process Manager configuration.xml
- Signavio Process Manager is configured in the Workflow Accelerator configuration file

7.2 Start Workflow Accelerator

After completing the installation as described above, both the MongoDB database and the Tomcat application server must be started for Workflow Accelerator to run properly. This section will give you more details on how to start them.

7.2.1 Windows

7.2.1.1 MongoDB

If you followed the instructions in this guide and registered a service for MongoDB, you can simply start the MongoDB server by opening a command line with administrator privileges and execute the following command:

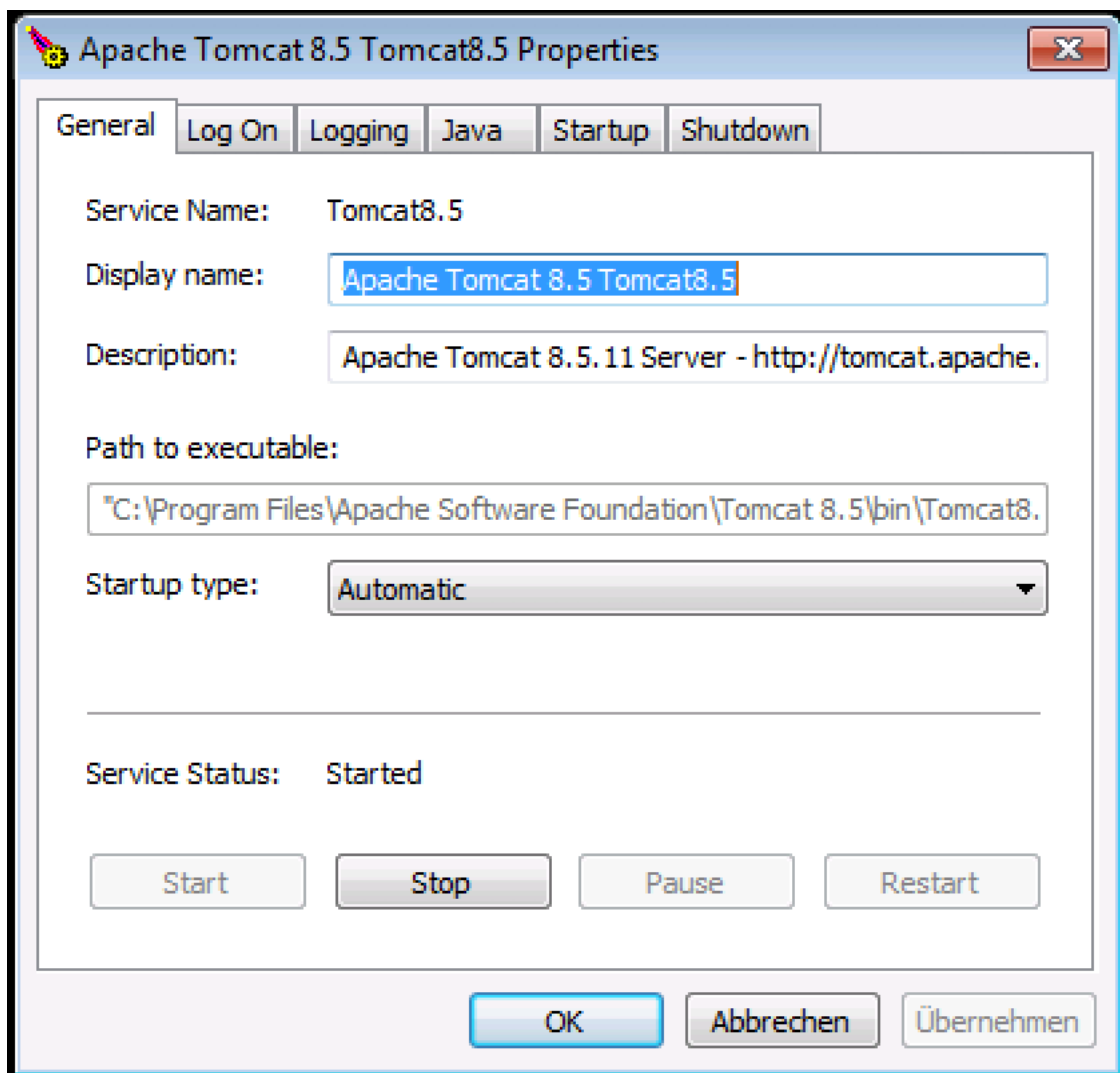
```
net start MongoDB
```

The registration also defined the startup type of the MongoDB service as “Automatic” which means it will start automatically if you restart the Windows system.

Alternatively, you can manage the MongoDB service by opening the `services.msc`. There, MongoDB will be listed among the other services and can be started and stopped.

7.2.1.2 Tomcat

Assuming you installed Tomcat using the Windows service installer, you can open the Tomcat configuration dialog again by opening the Windows start menu and selecting `Program Files\Apache Tomcat\Configure Tomcat`. The tab General contains respective buttons to start and stop the Tomcat service.



Furthermore, you have the option to define the startup type of the service. By default, this should be "Manual" which means you have to start the Tomcat server manually each time the server restarts. You can change the type to "Automatic". Then the Tomcat server will start automatically each time you restart your Windows server.

7.2.2 Script Engine

If you installed the script engine and wrapped it into a service, you can open the `services.msc`, locate the Signavio Script Engine service and start it there. Otherwise open a command line, navigate to the `$SCRIPT_ENGINE_HOME` directory and execute:

```
node server.js
```

Note, you cannot close the command line as long as the script engine is running.

7.2.3 Mail Relay

If you installed the mail relay and wrapped it into a service, you can open the `services.msc`, locate the Signavio Mail Relay service and start it there. Otherwise open a command line and execute (replace the placeholder with the absolute path):

```
java -jar $MAIL_RELAY_HOME\mail-relay.jar
```

Note, you cannot close the command line as long as the mail relay is running.

7.2.4 Debian

7.2.4.1 MongoDB

If you followed the instructions in this guide to setup MongoDB using a packet manager, some additional scripts have been installed that allow you to start, stop and restart the database very easily. Be aware, the execution of those scripts require root privileges. :

```
sudo service mongod start
```

If MongoDB is already running, you can restart the database by executing:

```
sudo service mongod restart
```

In order to determine whether MongoDB is running or not, you can check upon the status with the following command:

```
sudo service mongod status
```

Be aware, in earlier versions of MongoDB, e.g. 2.4.x, the scripts had a different name `/etc/init.d/mongodb`

```
/etc/init.d/mongodb start  
/etc/init.d/mongodb restart  
/etc/init.d/mongodb status
```

7.2.4.2 Tomcat

The recommended way to install Tomcat is to use a packet manager. Installing Tomcat under Debian with APT will install additional scripts to start, stop and restart the server. Be aware, the execution of those scripts might require root privileges:

```
/etc/init.d/tomcat8 start
```

You can restart the Tomcat server if it is already running by executing:

```
/etc/init.d/tomcat8 restart
```

Furthermore, you can check upon the status of the Tomcat and determine whether it is running or not with the following command:

```
/etc/init.d/tomcat8 status
```

7.2.4.3 Script Engine

If you installed the script engine and setup an Init script, execute the respective start command for the Init script. Otherwise open a command line, navigate to the `SCRIPT_ENGINE_HOME` directory and execute:

```
node server.js
```

Note, you cannot close the command line as long as the script engine is running.

7.2.4.4 Mail Relay

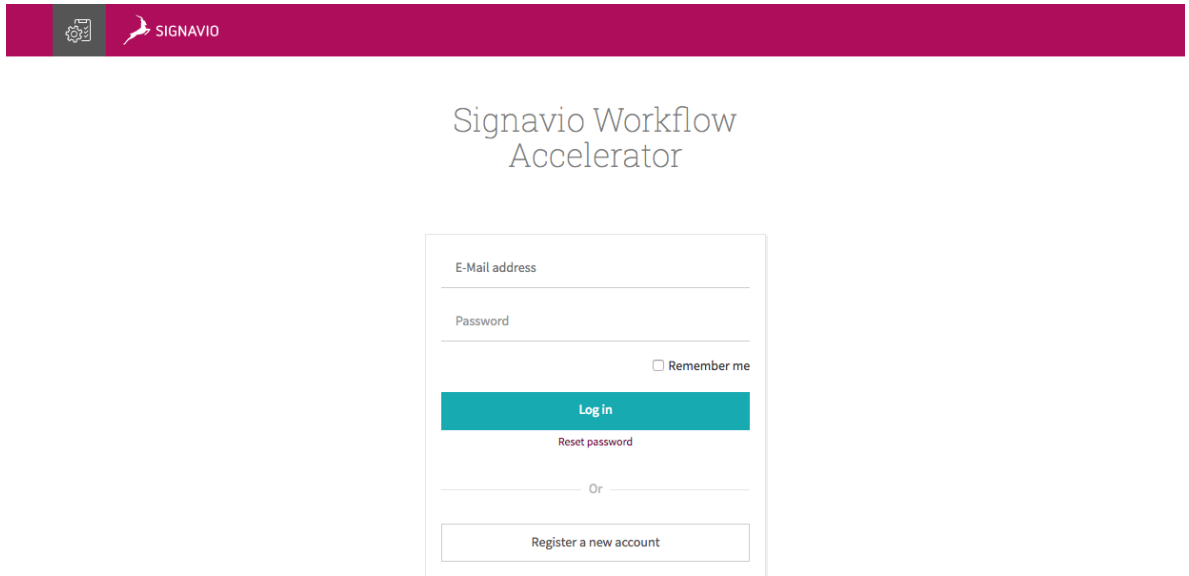
If you installed the mail relay and setup an Init script, execute the respective start command for the Init script. Otherwise open a command line and execute (replace the placeholder with the absolute path):

```
java -jar $MAIL_RELAY_HOME\mail-relay.jar
```

Note, you cannot close the command line as long as the mail relay is running.

7.3 Logging in to Workflow Accelerator

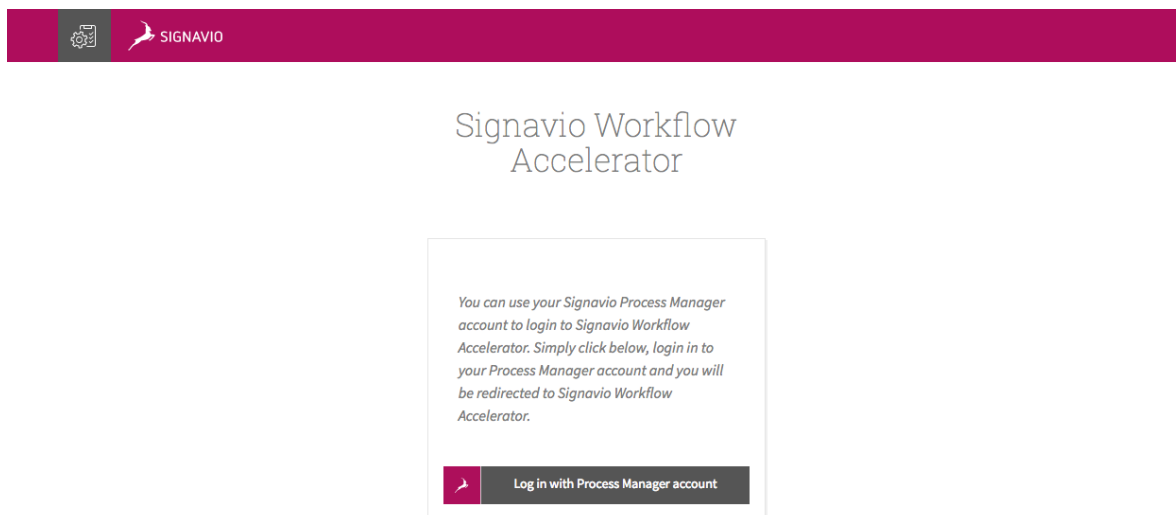
When the Workflow Accelerator system was started properly you can see the login screen by entering the URL of your Workflow Accelerator installation in your browser. There are different versions of the login screen. If you purchased an Workflow Accelerator standalone version, you will see a form that requires an email address and a password to login.



The screenshot shows the Signavio Workflow Accelerator login interface. At the top, there is a dark blue header with the Signavio logo on the left. Below the header, the text "Signavio Workflow Accelerator" is centered. The main content area is white and contains a login form. The form has two input fields: "E-Mail address" and "Password". Below the "Password" field is a checkbox labeled "Remember me". A prominent blue button labeled "Log in" is positioned below the "Remember me" checkbox. Below the "Log in" button is a link labeled "Reset password". Below the "Reset password" link is the text "Or". At the bottom of the form is a button labeled "Register a new account".

The default Workflow Accelerator login screen

If you purchased Workflow Accelerator along with Signavio Process Manager, you will see a login screen with one button. You can login to Workflow Accelerator using your Signavio Process Manager account as soon as the Workflow Accelerator users have been set up. See section **User management** for more information on how to set up the users.



The Signavio login screen

When you use the Process Manager integration with Workflow Accelerator to execute approval workflows, you can log in with your Workflow Accelerator credentials or with your Process Manager account.

The credentials for Process Manager and Workflow Accelerator are stored separately and are not copied or transferred between the two programs.

7.4 Create the initial user

In case you purchased Workflow Accelerator along with Signavio Process Manager and want to setup the integration, you don't have to create the initial Workflow Accelerator user manually. See section **Create the initial user** for more information.

Workflow Accelerator users are organized in organizations. There is a maximum amount of organizations that can be registered on an on-premise system. You will find the maximum number of available organizations in your license file. When the initial user registers, he will automatically create the first organization. In order to register, enter the URL of your Workflow Accelerator installation in your browser. You will see the login screen of Workflow Accelerator.

Click *Register a new account*, fill in your email address and click *Register a new user*. You will then receive an email with a link to complete your registration. After you have done that, your Workflow Accelerator organization is set up and you can go ahead and invite more users.

Every time a user registers using the register link on the login screen, a new organization is created until the maximum amount is reached. In general, you will create one organization and invite other users to join this organization. The next subsection explains how to invite users.

7.5 Invite other users

In case this is an integrated installation with Signavio Process Manager, see section [Signavio Process Manager integration](#) for setting up synchronization of users.

To invite other users, go to **Organization Settings** in the dropdown menu under your name in the top right corner. In **Invitations** you can enter email addresses of users you want to invite. The number of users that can be invited to your organization depends on the number of user licenses you have acquired. All organizations on an on-premise system share the same license pool. This means the shown number of unused licenses is the same in every organization. If one organization invites another user the count of unused licenses in all organizations decreases.

A user can be invited to multiple organizations. If a user is part of two organizations, he will require two user licenses, one for each organization he is part of.

Besides the manual invitation of every user, the enterprise version of Workflow Accelerator offers the possibility to synchronize users and groups with Active Directory. See [User and group synchronisation with Active Directory](#) for more information.

8 Stopping Signavio Workflow Accelerator

In order to stop the Workflow Accelerator system completely, you have to stop the Tomcat application server and the MongoDB database. However, if you need to restart the Workflow Accelerator system, it is sufficient to restart only the Tomcat server.

8.1 Windows

8.1.1 Tomcat

As already mentioned in the section **Starting Signavio Workflow Accelerator**, the Tomcat configuration dialog offers a simple way to stop the Tomcat server. Simply open the Windows start menu, select `Program Files\Apache Tomcat\Configure Tomcat`, open the tab *General* and click the button *Stop*.

8.1.2 MongoDB

If you have registered MongoDB as a Windows service, you can stop the database in two ways. Either you open the command line with administrator privileges and execute:

```
net stop MongoDB
```

Or you run `services.msc`, select MongoDB in the list of services and click on *Stop*.

8.1.3 Script Engine

If you installed the script engine and wrapped it into a service, you can open the `services.msc`, locate the Signavio Script Engine service and stop it there. If you are executing the script engine directly via command line, simply abort the process by pressing `CTRL + C`.

8.1.4 Mail Relay

If you installed the mail relay and wrapped it into a service, you can open the `services.msc`, locate the Signavio Mail Relay service and stop it there. If you are executing the mail relay directly via command line, simply abort the process by pressing `CTRL + C`.

8.2 Debian

8.2.1 Tomcat

If you installed Tomcat using a packet manager, you can simply run the following command. This might require root privileges.

```
/etc/init.d/tomcat8 stop
```

8.2.2 MongoDB

If MongoDB was setup following the instructions in this guide, you can simply stop the MongoDB server by executing the following command. This might require root privileges.

```
sudo service mongod stop
```

In earlier versions of MongoDB, e.g. 2.4.x, the script had a different name: `/etc/init.d/mongod`:

```
/etc/init.d/mongod stop
```

8.2.3 Script Engine

If you installed the script engine and setup an Init script, execute the respective stop command for the Init script. If you are executing the script engine directly via command line, simply abort the process by pressing `CTRL + C`.

8.2.4 Mail Relay

If you installed the mail relay and setup an Init script, execute the respective stop command for the Init script. If you are executing the mail relay directly via command line, simply abort the process by pressing **CTRL + C**.

9 Backup

In order to take a backup, it is sufficient to backup the content of MongoDB. All uploaded files are stored in the database and, therefore, covered by a database backup. MongoDB comes with built-in solutions for backups. Signavio recommends to use `mongodump`. For more detailed information about backups of MongoDB see the official documentation:

<https://docs.mongodb.com/manual/core/backups/>

<https://docs.mongodb.com/manual/tutorial/backup-and-restore-tools/#binary-bson-dumps>

Make sure the MongoDB server is running when you want to execute `mongodump`. If not stated otherwise via command line parameters, `mongodump` will try to backup data from the locally running `mongod` service.

You will need to use the Signavio Workflow Accelerator MongoDB user credentials again to access the data of the Workflow Accelerator database. If you are using another MongoDB user to perform the backups, make sure he has the necessary roles. For the backups, the user needs to access all databases and the user data. The user should have the following roles:

- `clusterAdmin`
- `userAdminAnyDatabase`
- `readWriteAnyDatabase`
- `dbAdminAnyDatabase`

You can specify the output folder where the content of the backup is stored. Be aware, if the directory contains already an older backup, the new backup will overwrite existing files. Alternatively, you could store several backups with a timestamp.

You should make sure the backup is stored in a safe location and cannot get lost if the server fails. Keep in mind, the backups can consume a lot of space, therefore, you should add additional disk space to the server if you plan storing on multiple versions of the backup.

The following lines will show an example on how to do a backup with `mongodump` on different operating systems.

9.1 Windows

The `mongodump.exe` can be found in the MongoDB installation directory, `$MONGO_HOME\bin\mongodump.exe`. You just have to create a directory where the backups should be stored and make sure the user doing the backups has the necessary read / write permissions. If you followed the instructions within this guide to setup MongoDB, the command to execute `mongodump` would look like this:

```
C:\MongoDB\bin\mongodump.exe -u signavio -p <SignavioUserPassword> -o C:\path\to\dumps
```

In the example above, adjust the location of your MongoDB installation accordingly. The parameter `-u` defines the username and `-p` the password of the Workflow Accelerator MongoDB user, replace them with your chosen values. The parameter `-o` specifies the output directory.

9.2 Debian

If you followed the instructions within this guide to setup MongoDB, then `mongodump` could be executed like this:

```
mongodump -u signavio -p <SignavioUserPassword> -o /path/to/dumps
```

The parameter `-u` defines the username and `-p` the password of the Workflow Accelerator MongoDB user, replace them with your chosen values. The parameter `-o` specifies the output directory, in this case `/path/to/dumps`. Make sure the executing user has the required privileges to write to the output folder.

Another example will create a folder with a timestamp for every new dump:

```
mongodump -u signavio -p <SignavioUserPassword> -o /path/to/dumps/`date +"%Y-%m-%d-%H-%M-%S"``
```

The folder `dumps` would then contain a subfolder for every backup with a name similar to “2014-10-07-17-49-07”. However, this method doesn’t remove old backups.

It's good practice to take backups regularly. We recommend using `cron` to schedule backups. You could for instance setup `/etc/crontab` that executes the `mongodump` command.

10 Restoring backups

Backups can be restored using MongoDB's `mongorestore` tool. Like `mongodump` you run it on the command line. For more detailed information, see the official `mongorestore` documentation:

<https://docs.mongodb.com/manual/reference/program/mongorestore/>

You will need to use the Signavio Workflow Accelerator MongoDB user credentials again to overwrite the contents of the Workflow Accelerator databases. We recommend running `mongorestore` with the parameter `--drop` which will remove every collection (table) from the database before it is restored.

10.1 Windows

The `mongorestore.exe` can be found in your MongoDB installation directory: `$MONGO_HOME\bin\mongorestore.exe`. If you followed the instructions in this guide to setup the database, you can execute the following command on the command line:

```
C:\MongoDB\bin\mongorestore.exe --drop -u signavio -p <SignavioUserPassword> C:\path\to\dumps
```

Replace the user credentials with the ones you have chosen during the setup and the directory with the one where you store your backups.

10.2 Debian

Using Debian, the command to restore a backup stored in `/path/to/dump` would look like this:

```
mongorestore --drop -u signavio -p <SignavioUserPassword> /path/to/dump
```

As you can see, you have to fill in the credentials of the Workflow Accelerator MongoDB user again. Furthermore, you need to specify where the backup is located.

11 Tooling

This section lists some tools that might help you to administer and maintain your Signavio Workflow Accelerator installation. Use them at your own discretion.

11.1 MongoDB

Robo 3T is a good tool for managing your MongoDB installation. It is free and available for different operating systems. Use it at your own discretion. Be careful when you use a tool to manipulate the database. Workflow Accelerator does not guarantee that any manipulated data can be restored. The tool can be downloaded from:

<https://robomongo.org/>

The download site offers two different applications. Robo 3T (former Robomongo) and Studio 3T a commercial tool for MongoDB, which offers an extended feature set.

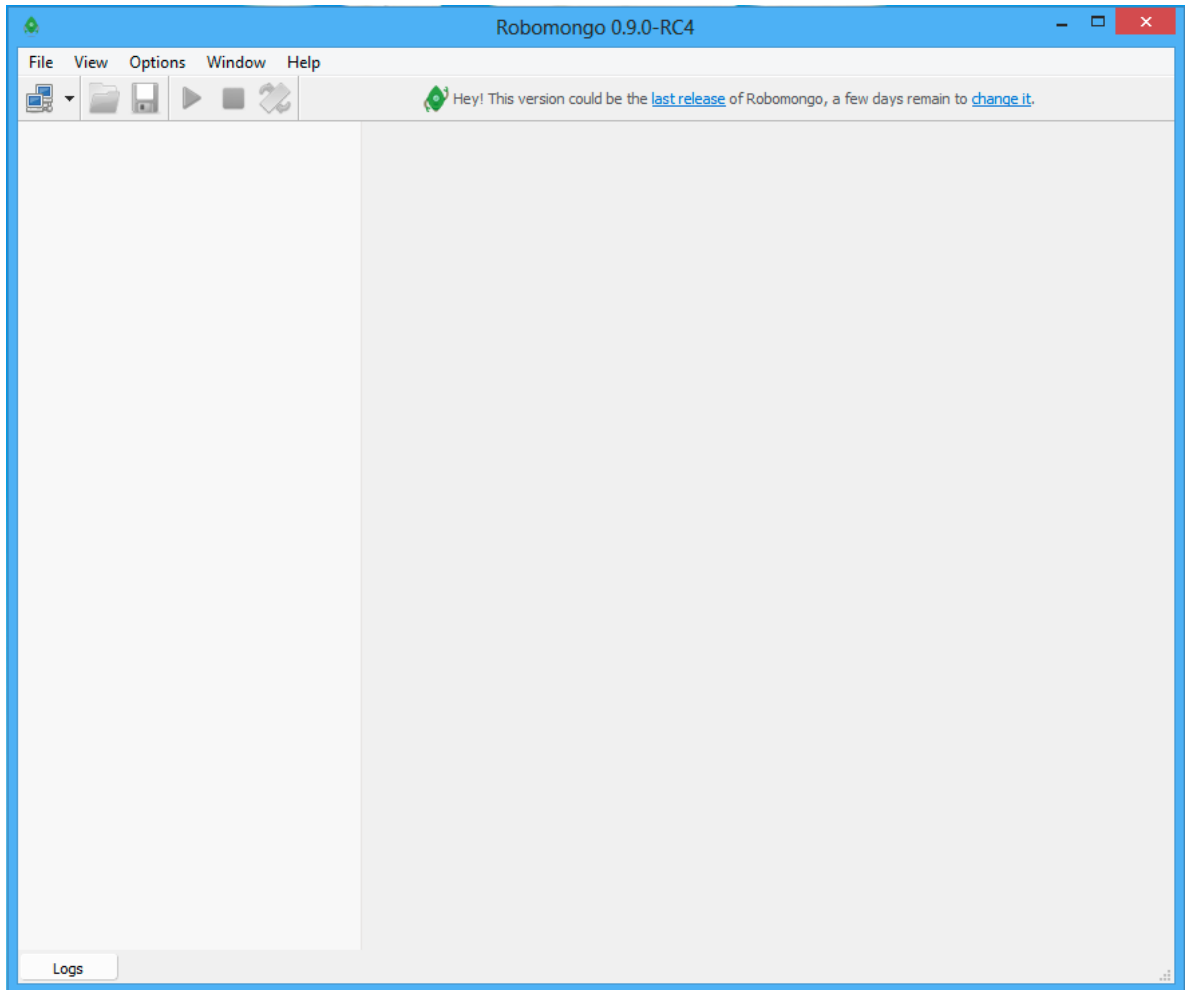
11.1.1 Connecting to MongoDB using Robo 3T

This section shows how to connect to your MongoDB server using Robo 3T. Note that MongoDB 3.2 requires you to use the latest version of Robo 3T.

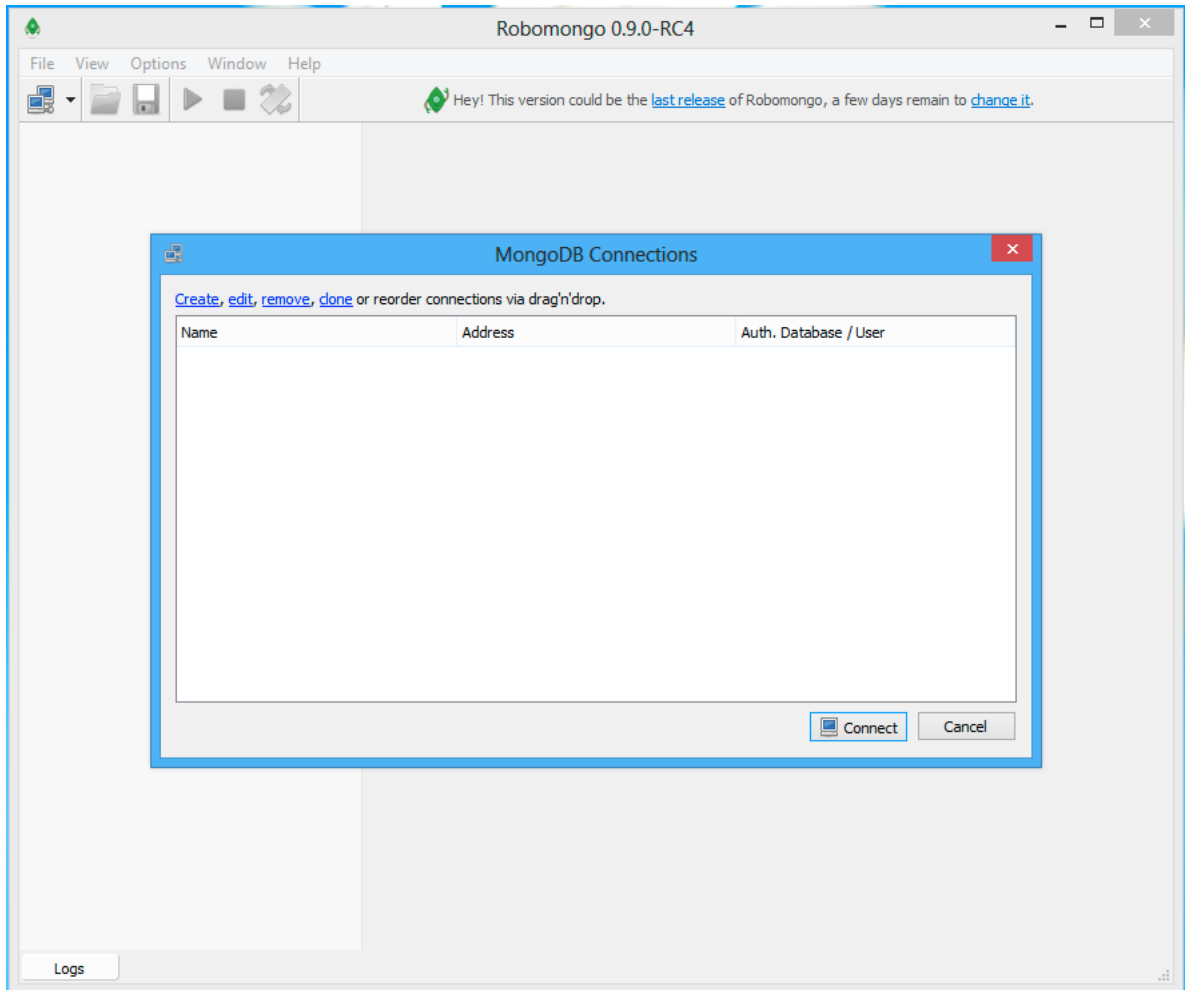
If you try to connect to your MongoDB from a remote system, ensure the MongoDB configuration has either no value defined for `bindIp` or lists the IP address you are trying to connect to.

Make sure your MongoDB server is running and open Robo 3T.

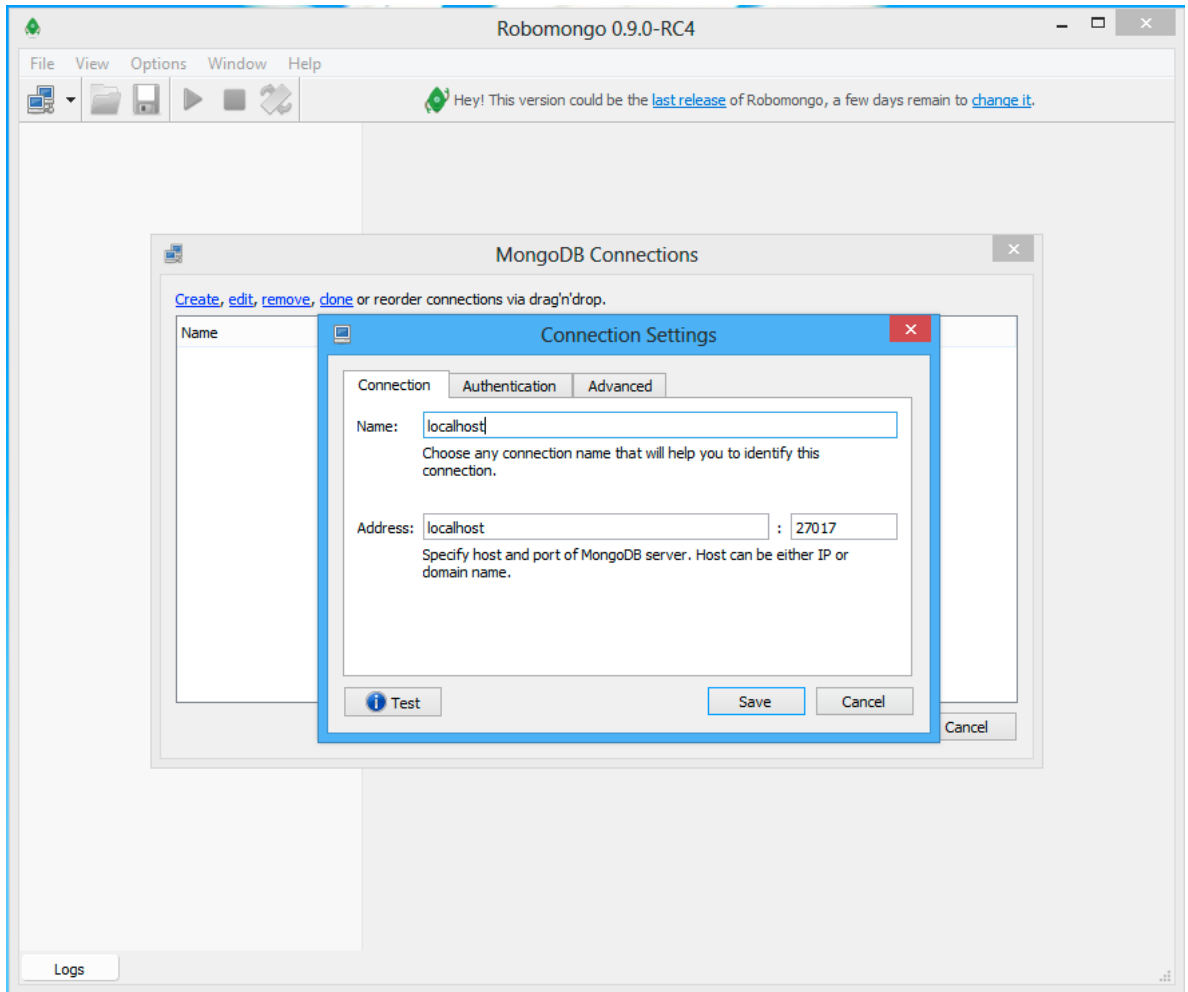
11 Tooling



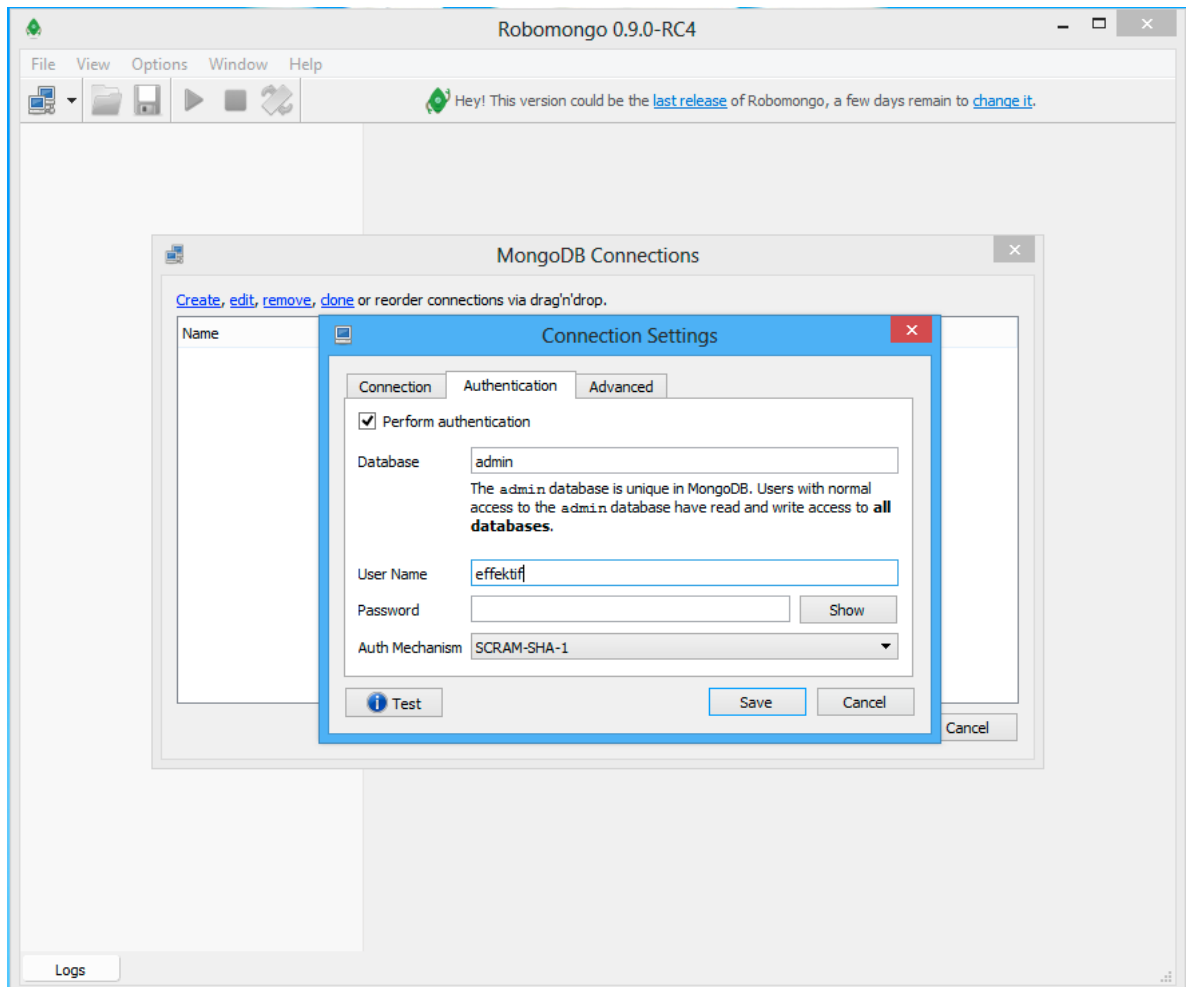
In the upper-left corner, click the icon to open a new connection.



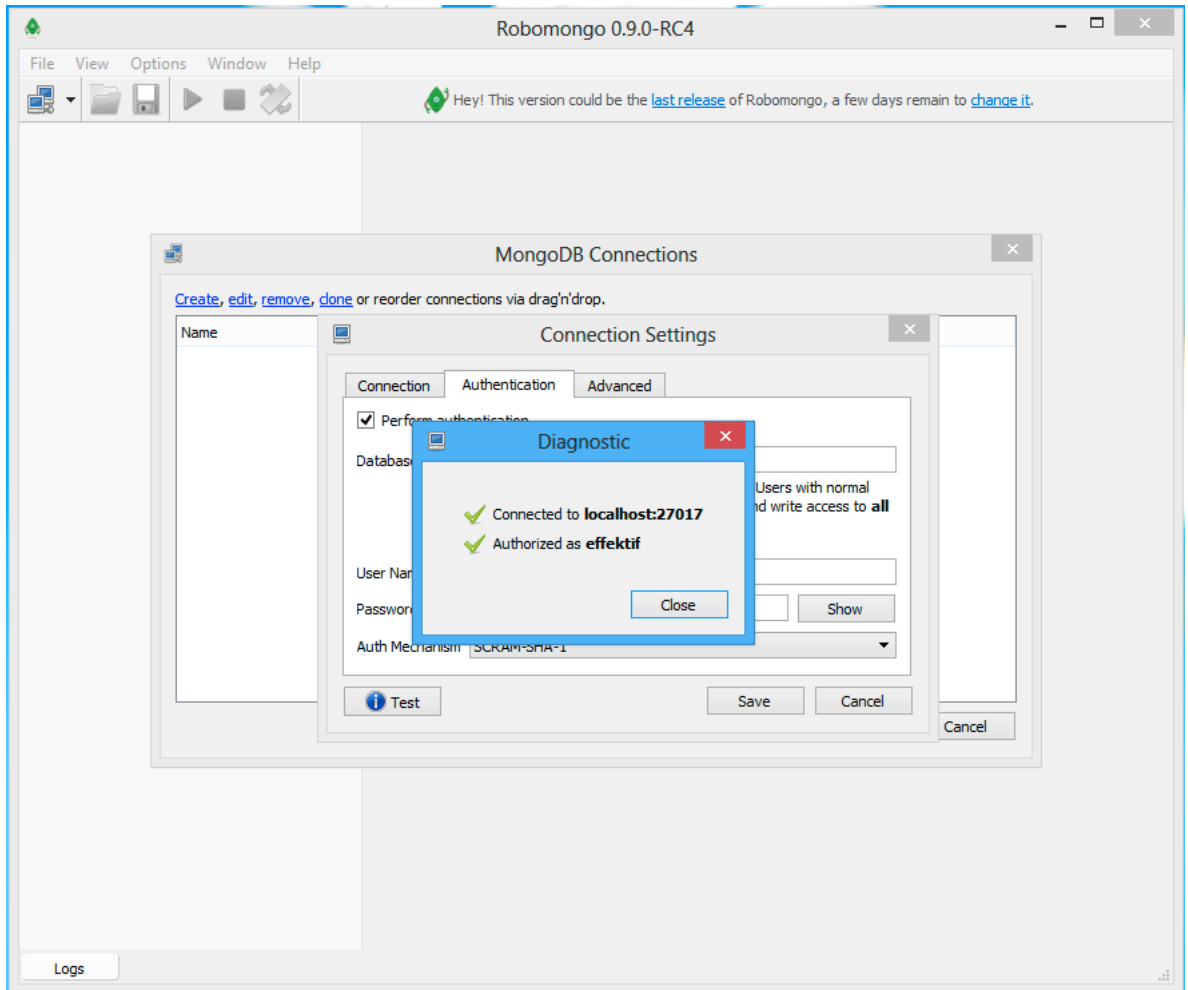
Click, the Create link to set-up a new connection.



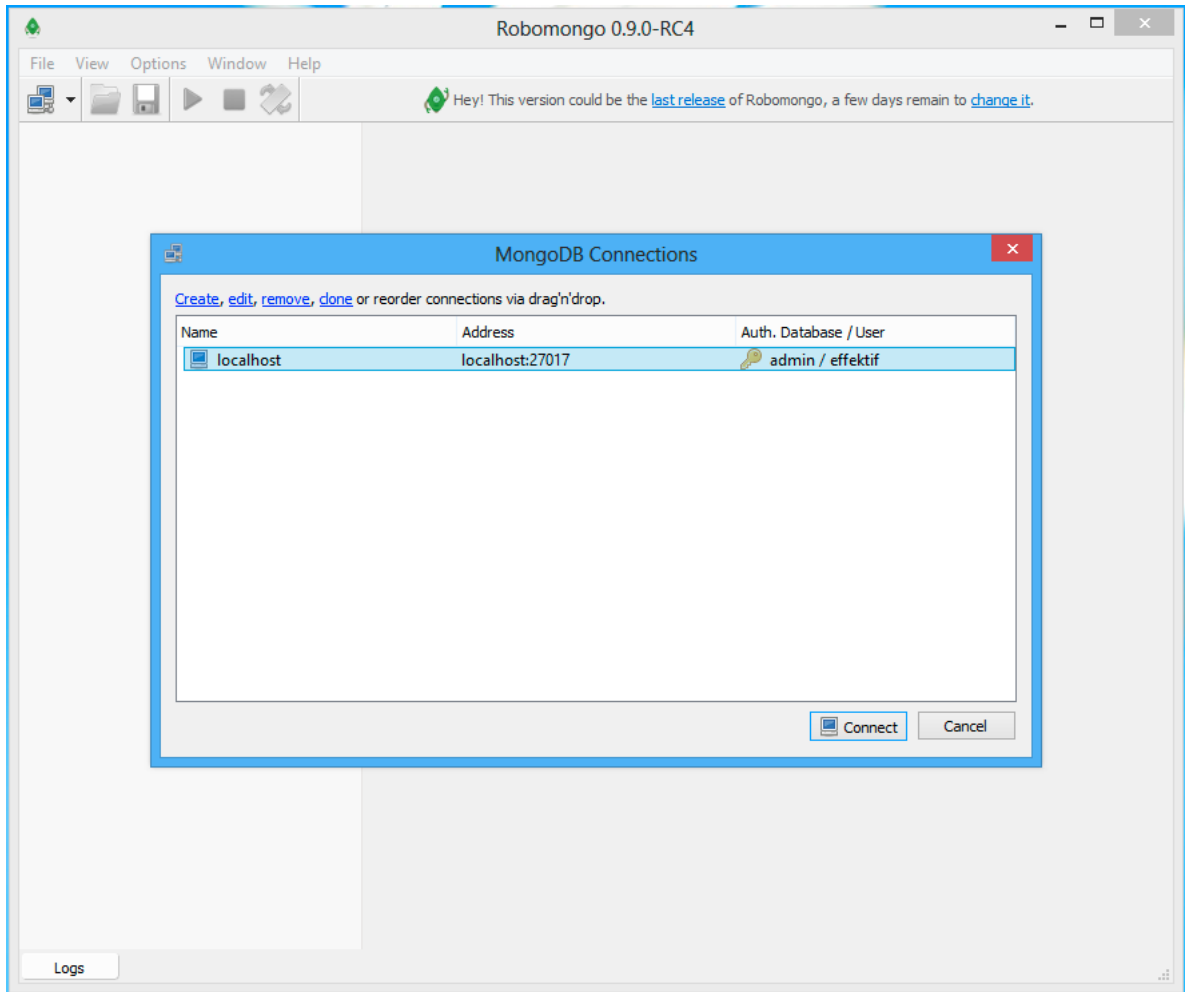
In the dialog you can rename the connection. Change the address if you are connecting to a remote server, otherwise leave localhost there.



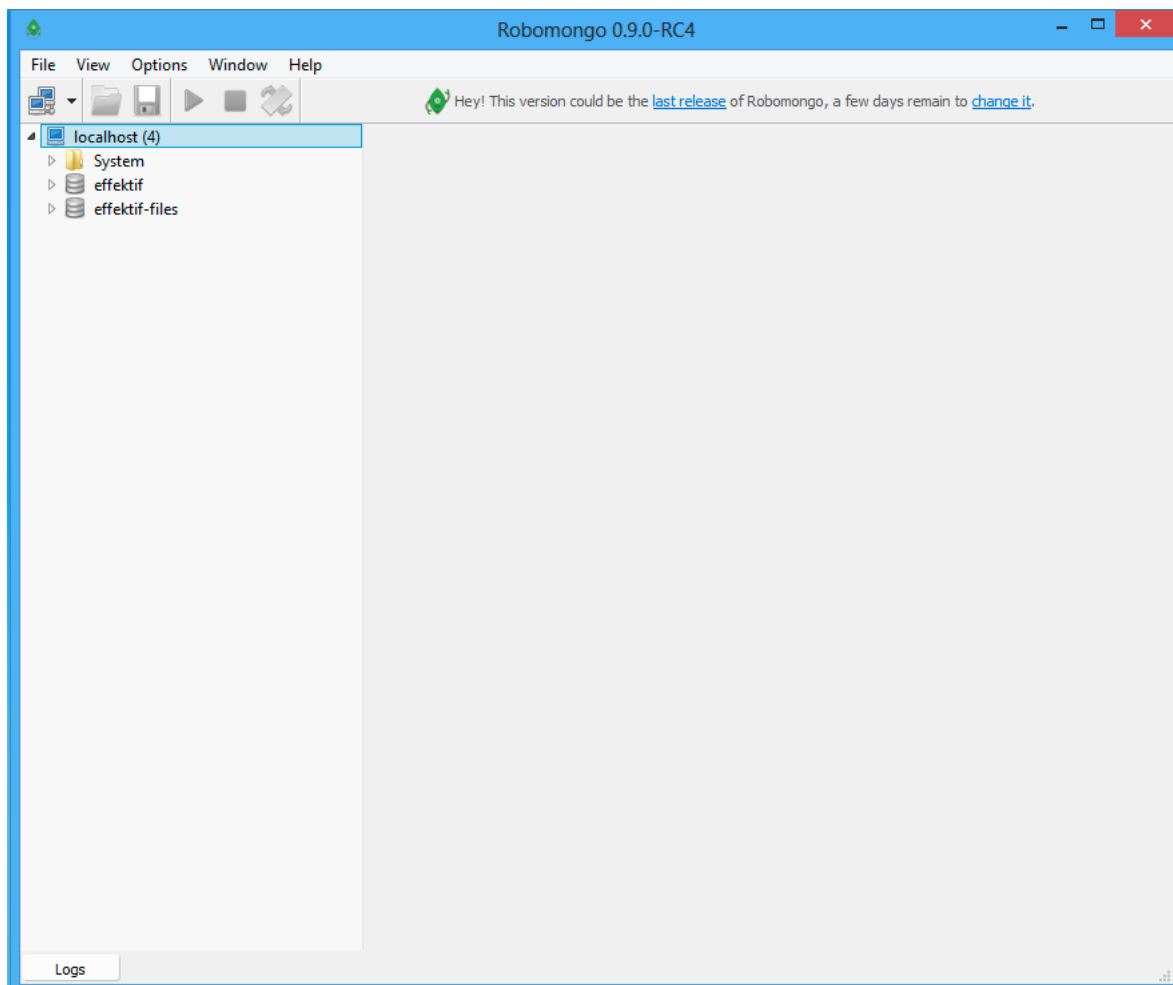
If you enabled authorization for your MongoDB server, select the Authentication tab and activate the Perform authentication option. Then enter the User Name (signavio) and the Password you have chosen before.



If you click Test, you should see a similar success message. Then click Save .



Select the new item in the list and click Connect .



You will see the connection in the left-hand panel and the two Workflow Accelerator databases below it.

11.1.2 Executing a JavaScript File for MongoDB

To execute a JavaScript file for MongoDB, the **Mongo Shell** is required (`mongo/mongo.exe`). The shell is already installed with most MongoDB installations and is available on the server on which MongoDB was installed.

To successfully execute a script, you need the database user and the name of the database from the `effektif-onpremise.properties` configuration file:

- Username - `effektif.mongodb.username`
- Password - `effektif.mongodb.password`
- Database name - `effektif.mongodb.database`

11.1.2.1 Windows

Execute the script:

1. Connect to the server running MongoDB.
2. Open the command line.
3. Replace the placeholders in `<>` and execute the following command.

```
mongo.exe <database name> --authenticationDatabase admin -u <user name> -p <password>  
<JavaScript-file>
```

If `mongo.exe` is not available in the path of your command line, you find the shell in the bin directory in the installation directory of your MongoDB.

On the command line you see the result of the execution and possible log outputs.

11.1.2.2 Linux

Execute the script:

1. Connect to the server running MongoDB.
2. Open the command line.
3. Replace the placeholders in `<>` and execute the following command.

```
mongo <database name> --authenticationDatabase admin -u <user name> -p <password>  
<JavaScript-file>
```

On the command line you see the result of the execution and and possible log outputs.

12 Troubleshooting

This section will assist you in finding the log files for every part of the Signavio Workflow Accelerator application. Additionally you find solutions for some common problems that can arise during the setup of the Workflow Accelerator system.

12.1 Where are the log files?

There are multiple different log files that yield information in case of an error scenario. The following subsections will assist you in finding the log files for every part of the Signavio Workflow Accelerator application. The most important one, regarding the Workflow Accelerator system is the `effektif.log`. It will contain any application related log entries.

12.1.1 Workflow Accelerator

The Workflow Accelerator log file `effektif.log` can be found in the location you configured in the logging configuration file `$(TOMCAT_HOME)/webapps/ROOT/WEB-INF/classes/logback.xml`. This file contains the line:

```
<property name="LOG_DIR" value="/path/to/log"/>
```

The value contains the directory of the log file. Besides the `effektif.log` the directory contains more log files for every day of the past 30 days the server was running. If you are looking for any information about incidents on a specific date, open the log file with the respective timestamp.

12.1.2 Script Engine

The Workflow Accelerator script engine log files have been configured in the configuration file `$(SCRIPT_ENGINE_HOME)/configuration.onpremise.js`. The values for the entries `logfile` and `logerrorFile` will tell you the name and location of the log files.

12.1.3 Mail Relay

The Workflow Accelerator mail relay log file `mail.log` has been configured in the logging configuration file `MAIL_RELAY_HOME/logback.xml`. This file contains the line:

```
<property name="LOG_DIR" value="/path/to/log"/>
```

The value contains the directory of the log file. Besides the `mail.log` the directory contains more log files for every day of the past 30 days the server was running. If you are looking for any information about incidents on a specific date, open the log file with the respective timestamp.

12.1.4 MongoDB

The MongoDB log file is configured by the `logpath` option. This is either defined in the MongoDB configuration file that you reference when starting MongoDB or directly as a start parameter of MongoDB. Anyway, the value of this parameter will tell you where to find the log file.

On Linux, one default location where the log file can be found is `/var/log/mongodb/mongodb.log`. Whereas default locations for the configuration file are `/etc/mongodb.conf` or `/etc/mongod.conf`. These locations can change depending on the distribution and depend on how you have installed MongoDB.

If you are using Windows and followed the instructions in this guide to setup MongoDB, you should find the log file in your MongoDB installation directory `$MONGO_HOME\logs`.

12.1.5 Tomcat

If not configured otherwise, the log files of the Apache Tomcat application server can be found either in the installation directory of the Tomcat `$TOMCAT_HOME\logs` (Windows) or a common log directory e.g. `/var/log/tomcat7` (Linux).

12.2 Common problems

Problem:	Solution:
<p>The Workflow Accelerator site shows a HTTP Status 500 error including the message <code>root cause java.lang.RuntimeException: Cannot find license.xml</code>.</p>	<p>You have not installed the Workflow Accelerator <code>license.xml</code> at all or placed it in the wrong directory. See Install the license file to solve the issue.</p>
<p>The Workflow Accelerator site shows a HTTP Status 500 error including the message <code>com/effektif/EffektifServlet : Unsupported major.minor version 51.0 (unable to load class com.effektif.EffektifServlet)</code>.</p>	<p>The Tomcat server is using the wrong Java version. Workflow Accelerator requires Java 8. Make sure you have installed the correct Java version. You can check your Java version by executing: <code>java -version</code> Installing Java explains how to install the correct Java version.</p>
<p>Executing a MongoDB command yields the error message <code>errmsg: "unauthorized"</code> or <code>errmsg: "auth fails"</code>.</p>	<p>Either the username / password combination that is used to execute the command is wrong or the respective MongoDB user lacks some required roles. The first issue can be checked by executing: Windows: <code>mongo.exe admin -u signavio -p <yourSignavioUserPassword></code></p> <p>Debian: <code>mongo admin -u signavio -p <yourSignavioUserPassword></code></p> <p>If this command fails with the error message <code>exception: login failed</code> you know that your username / password combination is wrong. Make sure you spelled the username and password correctly.</p> <p>To check the second issue, see Add a database user for Workflow Accelerator. This section lists all necessary roles and explains how to verify that the Workflow Accelerator user has all the required roles. Especially the restore and backup commands require certain roles to work properly.</p>

Problem:	Solution:
<p>Workflow Accelerator does not start and the log file contains the error message INFO LockingDbExecutor Database is locked by another executor. Wait...</p>	<p>The problem with the LockingDbExecutor arises when the application server is shut down during the initialization (update of licenses and migration of database) before the lock can be released again. You have to manually remove the lock from the database. We recommend using a tool like Robo 3T for this purpose. MongoDB explains how to install Robo 3T and set-up a connection to your MongoDB.</p> <ol style="list-style-type: none">1. Open Robo 3T.2. Establish a connection to MongoDB.3. Open the content of the Workflow Accelerator DB on the left side of the application.4. Under Collections you will find the table <code>properties</code>.5. A double click on <code>properties</code> shows all entries. In case of an error you will see three entries, if everything is normal there will be only two.6. Out of the three entries in the list, select the one that contains the value <code>schema.lock</code> for the key <code>k</code>.7. Use a right click to open the context menu for the respective entry and choose „Delete Document...“ to remove the entry. <p>IMPORTANT: Only the entry for <code>schema.lock</code> must be deleted. You are advised to create a backup of the database upfront.</p>