Signavio Process Manager Administrator's Guide

11.14.5

Contents

This document describes how to install and maintain the On Premise Edition of Signavio Process Manager on your server.

Chapter 1

Required administrator profile

As any web application, Signavio Process Manager depends on several external software components. The administrator guide requires prior knowledge about these components. The administrators responsible for installing and maintaining the Signavio Process Manager need to be proficient in the following technologies:

- The operating system you install the software on
- The database system you employ
- Oracle Java
- Apache Tomcat
- Setting up the connection to an email server
- If HTTPS is used, SSL/HTTPS
- If required, Microsoft Active Directory
- If required, Microsoft SharePoint Web Part

Important: Information for customers of the Signavio Small Business Edition

The Signavio Small Business Edition (available until spring 2013) has been provided with a pre-configured VMware image to simplify the installation of Signavio Process Manager. However, operating and maintaining the virtual machine is the responsibility of the customer. Due to the further development of Signavio Process Manager, the hardware requirements of the software have changed over the years. Therefore, the pre-configured VMware image no longer fulfills the requirements for operating Signavio Process Manager version 8.3.0 or later. If you are using the pre-configured VMware image and you want to update to version 8.3.0 or later, please migrate Signavio Process Manager on a server infrastructure as explained in this document.

Chapter 2

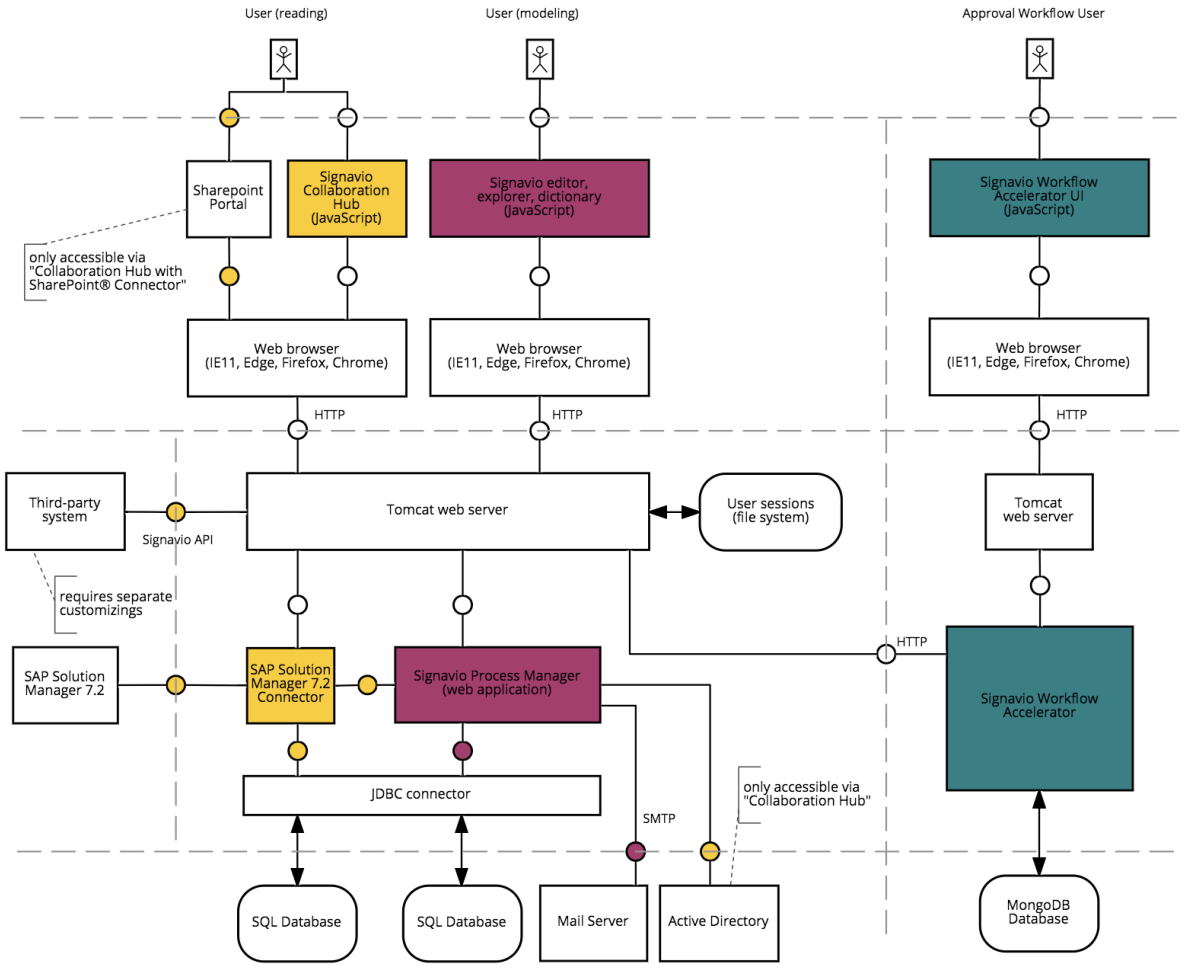
System architecture / components

This chapter provides an overview over the system architecture and briefly describes each software component.

2.1 System architecture

The following diagram describes the typical system architecture the Signavio Process Manager is embedded in together with third-party software such as web servers, web browsers and databases.

By default, the On Premise Edition contains the components highlighted in red, components or interfaces highlighted in yellow can be purchased separately.



System architecture overview

2.2 Software components

2.2.1 Signavio Collaboration Hub (JavaScript)

The front end of Collaboration Hub runs JavaScript and is entirely accessible through the web browser. It provides read-only access to published diagrams. Collaboration Hub grants read-only access to every user who is able to access the server of Signavio Process Manager. Collaboration Hub can be linked to an Active Directory service to enable single-sign-on authentication. Moreover, the active directory integration allows you to define detailed diagram access rights for Collaboration Hub users.

Signavio Collaboration Hub supports all popular web browsers (see: https://www.signavio.com/browser-compatibility/).

2.2.2 Signavio Process Manager (JavaScript)

The front end runs JavaScript and is entirely accessible through the web browser.

A user must be registered with a valid email address and a self-chosen password to get access to the Signavio Editor, Explorer or Dictionary.

2.2.3 Signavio Process Manager (web application)

The back end logic runs in the Tomcat runtime environment as a Java application.

2.2.4 JDBC Connector

Access to the database system is implemented with JDBC. An adapter must be downloaded from the website of the respective vendor (see section 3.7 Install JDBC driver).

2.2.5 Web browser

The Signavio front end is accessed via a web browser. The system requirements contain a list of all supported web browsers.

The front end requires a performant JavaScript engine and thus doesn't support Microsoft Internet Explorer 11 and lower.

2.2.6 Apache Tomcat

The application server represents the runtime environment for the Java back end. The system requirements contain a list of all Tomcat versions supported. For managing user sessions Signavio Process Manager uses the native Tomcat session mechanism. Such sessions are stored in the servers file system, so that they are kept even if the Tomcat application server is shut down. Therefore, a restart or a temporary unavailability of the Tomcat server is invisible to the user.

Furthermore, no constant connection to the backend is required for modeling in the frontend (except for saving, loading or dictionary lookups). This basically implies that the backend can be shut down while users are still modeling. If the Tomcat server is available again in time, no user will recognize it. Collaboration Hub is not accessible during a shut down.

HTTP or HTTPS (SSL) can be chosen for the connection between a web browser and the Tomcat server.

2.2.7 Mail server (SMTP)

The mail server can be any server that is accessible via SMTP.

2.2.8 Database system

The database contains Signavio Process Manager's user data. The data is stored using a UTF-8 character set. The system requirements contain a list of all supported database systems.

2.2.9 Active Directory

Signavio Collaboration Hub can be integrated with your Active Directory service. User and user group information stored in the Active Directory are used only to control read-only access to Collaboration Hub. Again, modeling access rights are decoupled from the Active Directory as they are managed from within Signavio Process Manager.

Important: When linking to an Active Directory using **Kerberos** as the authentication protocol, the Signavio server has to be installed on Windows 64 bit and must belong to the domain whose Active Directory needs to be accessed.

2.2.10 SharePoint portal

Optionally, Signavio offers a SharePoint Web Part as a WSP package. The package provides seamless front end integration for the read-only access to the SharePoint portal. The look and feel of Signavio Collaboration Hub can be customized easily via the Web Part.

2.2.11 Third-party system

Third-party systems (e.g. a ticket tracking system), can be connected with a separately licensed API.

2.2.12 Signavio Workflow Accelerator

Signavio Workflow Accelerator is a workflow automation system that is required for using the "approval workflow" feature. Signavio provides the software to customers who purchased a corresponding subscription. The back end is a Java web application that has to be installed on a separate server. Signavio Workflow Accelerator requires Oracle Java 7, Apache Tomcat 7 and MongoDB 2.4. Please refer to the Signavio Workflow Accelerator administrators guide at https://docs.signavio.com/adminguide/workflow/en/ for more information.

2.2.13 Signavio Workflow Accelerator UI

The user interface of Signavio Workflow Accelerator is accessed through a web browser.

Chapter 3

Installation

The following chapter provides step-by-step instructions on how to install Signavio Process Manager. You need to have the ZIP archive available that contains Signavio Process Manager's installation files, as well as a license file.

Note: In the installation instructions, the term `$TOMCAT_DIR` is used in command line statements to denote the path to the Apache Tomcat instance that runs the Signavio Process Manager web application. To be able run the commands right away after copying & pasting them, you can set a corresponding environment environment variable.

3.1 System requirements

Signavio Process Manager is a client-server application and requires at minimum two servers - a **web server** and a **database server** - that may run on the same machine.

3.1.1 Hardware minimum requirements for the web server

The following hardware is required for Signavio Process Manager's web server:

- 2 CPU cores (virtualization possible)
- Support of 64 bit systems
- 4 GB RAM (dedicated to the application server)
- 20 GB hard-drive space

3.1.2 Hardware minimum requirements for the database server

The following hardware is required for Signavio Process Manager's database server:

- 2 CPU cores (virtualization possible)
- Support of 64 bit systems
- 4 GB RAM (dedicated to the database)
- 30 GB hard-drive space (enough for approx. 20,000 diagram revisions when using a MySQL database)

Of course it is possible to run the web server and the database server on one logical server. The hardware minimum requirements accumulate in this case.

When using an Oracle database, the minimum hard-drive space will be approx. 1.5 times higher, so that 20000 diagram revisions will require at least 45 GB.

3.1.3 Software requirements for the web server

The following software is required for Signavio Process Manager's web server:

- Operation system:
  - Debian Stable Release 64-Bit
  - Microsoft Windows 64-Bit
- Oracle Java 8 (64Bit)
- Apache Tomcat 8.X.

64 bit is required so that the Java VM is able to address more than 1.5 GB of heap space.

Moreover, you need to connect Signavio Process Manager to an email server. Please make sure the email server can be reached within your network and create an account for Signavio Process Manager or allow anonymous access.

Note: In case you have purchased Collaboration Hub and want to use Kerberos authentication for single-sign-on, please make sure the web server connecting to your Active Directory is running on Microsoft Windows 64 bit.

3.1.4 Software requirements for the database server

The following software is required for Signavio Process Manager's database server:

- Operation system:
  - Debian Stable Release 64Bit
  - Microsoft Windows 64Bit
- Database:
  - Recommended: MySQL Standard Edition 5.x (InnoDB Storage Engine)

    or

    MySQL Enterprise Edition 5.x (InnoDB Storage Engine)
  - Oracle 11g
  - Oracle 12
  - MS SQL Server 2008
  - MS SQL Server 2012
  - MS SQL Server 2014
  - MS SQL Server 2016

3.1.5 Hardware minimum requirements for the client

- Processor: 2,00 GHz or more
- 4 GB RAM
- 1 GB empty disk storage

3.1.6 Software requirements for the client

- Operation System:
  - Microsoft Windows
  - Mac OS X
  - Linux
- Web Browser
  - Microsoft Edge
  - Microsoft Internet Explorer 11
  - Mozilla Firefox from version 52
  - Apple Safari from version 10
  - Google Chrome from version 50

3.1.7 System requirements for Signavio Workflow Accelerator (optional)

You need to set up a Signavio Workflow Accelerator server to be able to use the **approval workflow** feature of the Enterprise Edition. Signavio Workflow Accelerator has to be installed on a separate server. Please refer to the Signavio Workflow Accelerator administrators guide[1] for more information about the system requirements and the installation process.

3.2 Configuring the operating system (web server & database server)

In case you are using a Linux distribution, please ensure that your operating system is using UTF-8 as the default encoding method. Detailed instructions on how this is configured can be found in the manual of your operating system.

3.3 Configuring the network

You can define a DNS record for your Signavio system in your network. Please make sure the IP address or the DNS record is set in the configuration of the Signavio Process Explorer.

3.4 Installing Java (web server)

Install Java 8 64bit on the web server.

You can download the installation package here:

http://www.java.com/en/download/manual.jsp

[1] https://docs.signavio.com/adminguide/workflow/en/

Important: Please pay attention to version number. Signavio Process Manager doesn't support Java 9, yet.

3.5 Configuring the web server

3.5.1 Installing Apache Tomcat

You can download the Apache Tomcat installation files for different operating systems at:

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

Some Linux distributions allow the installation of Apache Tomcat through the package manager. If you use Debian version 6.0 or higher, for example, you can install Tomcat as follows:

```
apt-get install tomcat8
```

For Windows, Apache provides the Windows Service Installer (download the package `32-bit/64-bit Windows Service Installer`).

This allows installing the Apache Tomcat as a windows service by using the Apache Tomcat installation wizard.

Note: After installing Tomcat through the Windows wizard, it will start up automatically at every server startup.

When using the Windows installer, some properties can be easily configured via the administrator's panel (see below). Therefore, we recommend the Service Installer to be used on Windows systems. Please make sure you run the Service installer with administrator rights.

Important: Please make sure the executing user has got read/write permissions for the folder `$TOMCAT_DIR/webapps` and all containing sub folders.

On a Windows system, we recommend installing Apache Tomcat into a root folder on `C:\`. That should make sure that the executing user has got read/write permissions for the respective folders.

3.5.2 Configuring Apache Tomcat

Allocate as much RAM as possible for Apache Tomcat. We recommend to allocate at least 4096MB RAM.

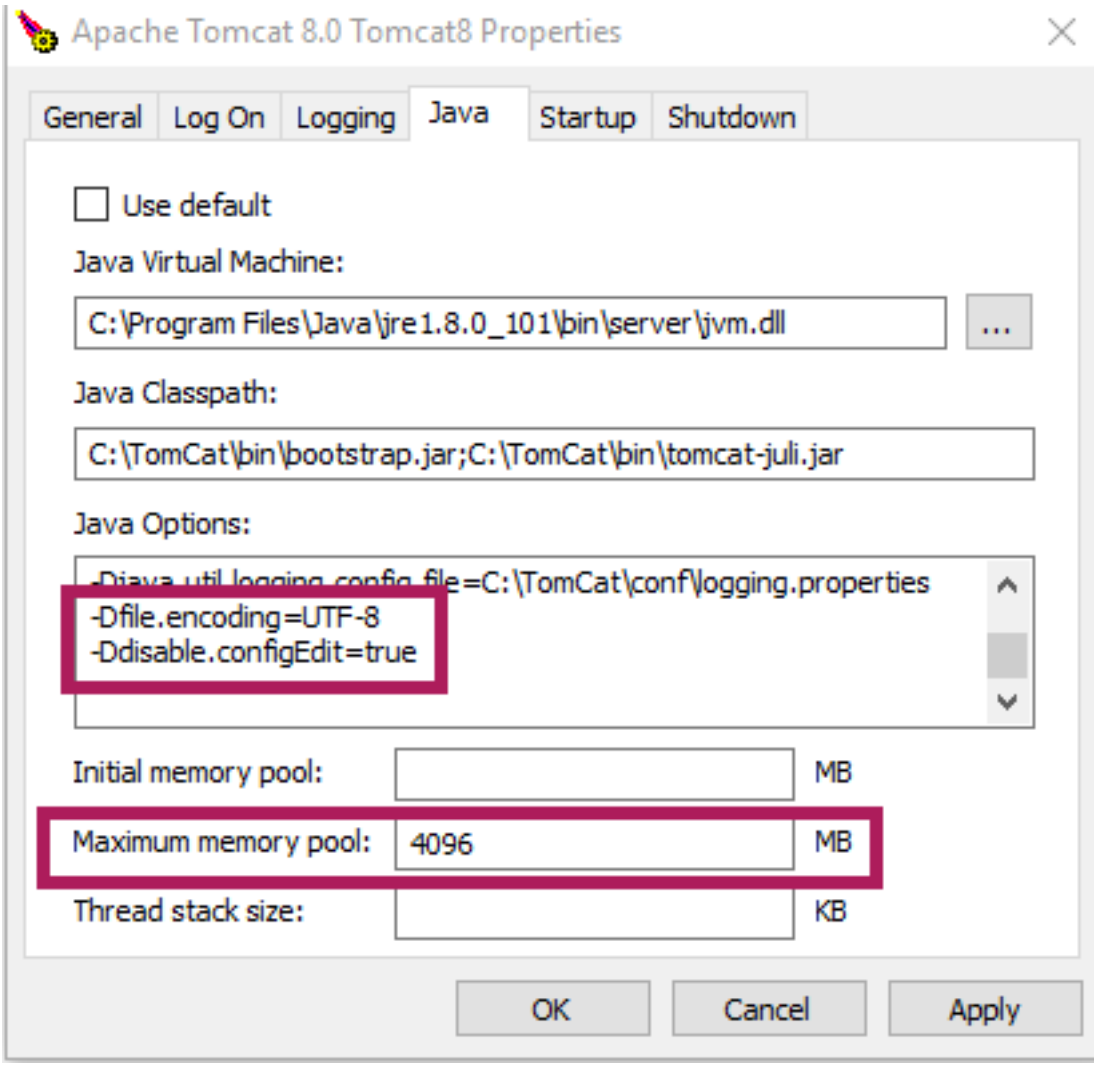
This is done by changing the JVM variable `Xmx` (that means executing the JVM start command with the attached parameter `-Xmx4096m`). When using the Apache Tomcat installer on a Windows system, you can edit JVM variables in the Apache configuration dialog.

Please ensure Apache Tomcat is using `UTF-8` as `default encoding` mechanism and has configuration editing disabled. Set the JVM variable `Dfile.encoding` to `UTF-8` and `-Ddisable.configEdit` to `true`:

```
-Dfile.encoding=UTF-8
-Ddisable.configEdit=true
```

On Windows, open the Windows Start menu and go to **Program Files** - **Apache Tomcat X** - **Configure Tomcat**.

Go to the tab **Java** and enter the values as follows:



Configure Apache Tomcat

On Linux, execute the following command:

```
export JAVA_OPTS="$JAVA_OPTS -Xmx4096m -Dfile.encoding=UTF-8 -Ddisable.configEdit=true"
```

Please ensure the XML file `$TOMCAT_DIR/conf/web.xml` is encoded with `UTF-8` and change the configuration if necessary.

The first line of the file has to be:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Additionally, add the attribute `URIEncoding="UTF-8"` to the `Connector` element of the file `$TOMCAT_DIR/conf/server.xml`.

To compress the information transferred, automatic compression should be enabled.

Add the following attributes to the `Connector` element:

```
compressableMimeType="
        text/css,text/plain,
        image/svg+xml,
        application/xhtml+xml,
        text/html,
        text/xml,
```

```
        text/javascript,
        application/xml,
        application/x-javascript,
        application/javascript"

compression="on"

compressionMinSize="2048"
```

Moreover, you can configure the port the Apache Tomcat is using by setting the attribute `port` of the `Connector` element. By default, the port is `8080`. If you want to spare users of Signavio Process Manager entering the port number to the address line of the web browser (e.g. `http://signavio.yourCompany.com:8080/`), change the port number to `80`.

To support diagrams that exceed 2 MB of mere textual representation, you can add the following line to the connector definition:

```
maxPostSize="33554432"
```

Example of a fully configured connector:

```
    <Connector port="80" protocol="HTTP/1.1"

               connectionTimeout="20000"

               redirectPort="8443"

               URIEncoding="UTF-8"

               compressableMimeType="
               text/css,text/plain,
               image/svg+xml,
               application/xhtml+xml,
               text/html,
               text/xml,
               text/javascript,
               application/xml,
               application/x-javascript,
               application/javascript"

               compression="on"

               compressionMinSize="2048"
maxPostSize="33554432"
    />
```

Note:

- Signavio Process Manager uses the `webapps` directory `ROOT`. This may cause trouble if other applications are running on the same Tomcat server. Therefore, we strongly recommend you to exclusively use the Tomcat server for Signavio Process Manager.
- If you want to use HTTPS, please ensure that the certificate is signed by a trusted certificate authority (CA) and that the JVM trusts the Tomcat's certificate. You can find further information on this topic in the chapter `3.20 Using HTTPS (optional)`.
- If you use multiple connectors, apply the correct settings to all of them.
- If you want to integrate the Tomcat server into your system via the AJP connector, please make sure the connector is UTF-8 encoded.

```
<Connector port="8089" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8" />
```

3.6 Configuring the database server

Important: The database system should be installed by an experienced database administrator. The various database systems differ in terms of installation and configuration and will not be explained in detail in this manual.

This section describes how to configure the database server.

Hint: We recommend you use MySQL Standard Edition 5.x (InnoDB Storage Engine) or MySQL Enterprise Edition 5.x (InnoDB Storage Engine), as long as your organization doesn't specifically require you to use a different database system.

The database system is required to support **transactions**.

If you are running a database cluster in your company, a database for the Signavio System can be created there, too.

A user account for Signavio Process Manager has to be created in the database.

Apart from the standard SQL commands `SELECT`, `UPDATE`, `INSERT` and `DELETE`, Signavio Process Manager user must be able to `create`, `edit` and `delete` `tables` `views` and `indexes` and to set `locks`.

`Routines` (stored procedures) are not created by the Signavio Process Manager account. Moreover, the system doesn't change any database user account settings.

To use the *Diagnostics tool for system administrators* (page 27), your database user needs to be able to create views. You can configure this right with the following command (all supported database systems):

```
GRANT CREATE VIEW ON <db_schema> TO <dbuser>
```

, with `<dbuser>` being a placeholder for your database user and `<db_schema>` being a placeholder for your database schema name.

Furthermore, you need to grant the database user permission to read out information from a set of meta data tables:

Oracle

```
GRANT SELECT ON user_tab_columns TO <dbuser>
GRANT SELECT ON user_tables TO <dbuser>
GRANT SELECT ON user_ind_columns TO <dbuser>
GRANT SELECT ON user_constraints TO <dbuser>
GRANT SELECT ON user_cons_columns TO <dbuser>
```

MySQL

```
GRANT SELECT ON information_schema.columns TO <dbuser>
GRANT SELECT ON information_schema.tables TO <dbuser>
GRANT SELECT ON information_schema.statistics TO <dbuser>
GRANT SELECT ON information_schema.table_constraints TO <dbuser>
GRANT SELECT ON information_schema.key_column_usage TO <dbuser>
```

SQL Server

```
GRANT SELECT ON information_schema.columns TO <dbuser>
GRANT SELECT ON sys.dm_db_partition_stats TO <dbuser>
GRANT SELECT ON sys.objects TO <dbuser>
GRANT SELECT ON sys.indexes TO <dbuser>
GRANT SELECT ON sys.index_columns TO <dbuser>
GRANT SELECT ON sys.foreign_keys TO <dbuser>
GRANT SELECT ON sys.foreign_key_columns TO <dbuser>
```

with <dbuser> being a placeholder for your database user.

Signavio Process Manager is linked to the database server via a JDBC driver. Please ensure that the database is accessible for the Signavio server through your network.

Hint: You need the JDBC-URL of the database to configure it. Please request such an URL from the responsible database administrator.

3.6.1 Configuring MySQL

Important: Only required when using a MySQL database

Please ensure that by default the MySQL database is encoded with `UTF-8` and uses the `InnoDB` storage engine. This is the only storage engine delivered by MySQL that supports transactions. Additionally, the maximum package size must be set to 16 MB.

Hint: We recommend to also install the MySQL Workbench. The administration of the MySQL server is much easier with it.

Those configurations are made by changing the following variables in the MySQL configuration file (`my.ini` or `my.cnf`).

The file should be found in the respective folder (e.g. `C:\ProgramData \MySQL\MySQL Server 5.5` on Windows and `/etc/mysql` on Debian-based Linux distributions).

Use the following configuration, if you are running **MySQL up to version 5.5.2**:

```
[mysqld]

max_allowed_packet=16M

default-character-set=utf8

default-collation=utf8_general_ci

default-storage-engine=innodb

max_connections=160
```

Use the following configuration, if you are running **MySQL version 5.5.3 or later**:

```
[mysql]

default-character-set=utf8

[mysqld]

max_allowed_packet=16M
```

```
character-set-server=utf8

collation-server=utf8_general_ci

default-storage-engine=INNODB

max_connections=160

query_cache_limit = 32M

query_cache_size = 128M

query_cache_type = 1
```

Create an empty database scheme for Signavio Process Manager.

Hint: Ensure to overwrite older entries or make sure that no older entries are below their new corresponding ones. Replace if necessary, or use # to comment the lines out.

Hint: If you can't locate the configuration files, you can also edit this attributes in the MySQL Workbench.

3.6.2 Configuring Microsoft SQL Server

Important: Only required when using a Microsoft SQL database

If you are not hosting the MSSQL Server on MS Windows, please ensure that the host system supports `UTF-8` encoding.

Moreover, the TCP/IP protocol must be enabled for the database instance. To enable this, start the **SQL Server Configuration Manager** and selecting **SQL-Server-Network configuration** - **Protocols for 'INSTANCE_NAME'** in the list and activate the TCP/IP protocol. After this configuration the SQL Server must be restarted.

Create a database user account which uses SQL Server authentication. The user name and password are required to configure the Signavio Process Manager later on. Please check, if the option **SQL Server- and Windows-authentication mode** is enabled in the security settings of your Microsoft SQL Server instance.

Create an empty database for the Signavio Process Explorer and assign the user account the **membership** `db_owner` for the database. Set `dbo` as the `default scheme` for the user account. Use `Latin1_General_CI_AS` as the database's *collation*.

3.6.3 Configuring Oracle database

Important: Only required when using an Oracle database

Please ensure the database is using `UFT-8` as standard encoding and create an empty database for Signavio Process Manager.

When using a Linux operating system you find the configuration file by default at `oracle/network/admin/sqlnet.ora`.

3.7 Installing the JDBC driver

To use the database system for the Signavio application server, add a compatible JDBC driver to the Apache Tomcat installation. You can download the JDBC driver from your database vendor's support homepage, as described in the following sections. Please consider the vendor's license agreements for using the JDBC driver.

Hint: Please ensure you install exactly the JDBC driver version that is mentioned in the corresponding section for your database system. Other versions of the JDBC driver (older and newer) may cause problems.

After you have successfully downloaded the JDBC driver, copy the jar library of the JDBC driver in the Apache Tomcat installation directory (below called `$TOMCAT_DIR`) to the folder **$TOMCAT_DIR/lib**.

3.7.1 Downloading the JDBC driver for MySQL

Please download the MySQL Connector/J JDBC driver **version 5.1.45** here:

Linux:

http://dev.mydql.com/get/Downloads/Connector-J/mysqul-connector-java-5.1.45.tar.gz

Windows:

http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.45.zip

Alternatively, open the download website of the MySQL Connector/JDBC driver in your web browser:

http://dev.mysql.com/downloads/connector/j/

Please ensure to select version 5.1.45 and as platform **Platform Independent**.

Unpack the archive to a location of your choice. The JDBC driver is the file `mysql-connector-java-5.1.45-bin.jar`.

3.7.2 Download JDBC driver for Microsoft SQL Server

Please go to the download website of Microsoft to get the Microsoft JDBC driver **version 6.0**:

https://www.microsoft.com/download/details.aspx?id=11774

Click the **Download** button and select the corresponding file (Windows: `enu\sqljdbc_6.0.8112.100_enu.exe`, Linux: `enu\sqljdbc_6.0.8112.100_enu.tar.gz`) and click **Next**. Unpack the self-extracting archive to a location of your choice. The correct JDBC driver is the file `sqljdbc42.jar` in the `$TOMCAT_DIR/libs` folder. `$TOMCAT_DIR` denotes the root directory of your Tomcat instance.

3.7.3 Download JDBC driver for Oracle

For downloading the JDBC driver an account on oracle.com is required. The account is available to you as a customer of Oracle. Independent of the Oracle database version you are using, please download the Oracle Database JDBC driver version 12.1.0.1 from the Oracle download website:

http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html

First, read and accept Oracle's license agreement (OTN Development and Distribution License Agreement). Subsequently, download the files `ojdbc7.jar` and `orai18n.jar`. If you haven't logged in before, you will be requested to log in before the download starts.

The JDBC driver consists of the two files `ojdbc7.jar` and `orai18n.jar`.

3.8 Installing Signavio Process Manager

Installing the program files for Signavio Process Manager requires only a few steps.

1. Look for the directory webapps in the Tomcat installation directory (we'll refer to it as `$TOMCAT_DIR`). This directory contains the web applications' program files that have to be executed by Apache Tomcat. We recommend and assume that for the initial installation of Signavio Process Manager a newly installed Apache Tomcat is used, with no other web applications running.
2. Stop the Apache Tomcat (if running).
3. Delete all files from the directory `$TOMCAT_DIR/webapps`.
4. Delete the directory `$TOMCAT_DIR/conf/Catalina`.
5. Delete the directory `$TOMCAT_DIR/work/Catalina`.
6. Unpack the ZIP archive containing the installation files.
7. Copy the whole content of the directory `files` that contains the installation files into the folder $TOMCAT_DIR/webapps. Copy **only the content of the directory** but not the directory itself!

3.9 Configuring SAP Solution Manager Connector

In order to use the SAP Solution Manager 7.2 (SolMan) connector, you must first install Apache Tomcat and Signavio Process Manager. Refer to the *Installation* (page 9) chapter for instructions.

Note: You must run the connector on a different database schema than the one you use for Signavio Process Manager.

3.9.1 Installing the connector

To install the SAP Solution Manager connector, you need two files: `solman72.yml` and `configuration.xml`. `solman72.yml` is used to configure the SAP Solution Manager connector, while `configuration.xml` is the same file as the one used to configure Signavio Process Manager. Both are provided in the .zip file that comes with the on-premise edition of Signavio Process Manager.

First, copy `solman72.yml` into the Tomcat `conf` folder. Open the `solman72.yml` file, and look for the following section:

```
signavio:
  client:
    # the shared secret for communication between Signavio Process Manager and solman72-service
    # must be the same as the <sharedJwtSecret> in configuration.xml
    sharedJwtSecret: SHARED_SECRET_CHANGE_ME
  platform:
    # the URL of the Signavio Process Manager
    # must be the same as the <server> in configuration.xml without a slash at the end
    url: http://localhost:8080
  resource:
    userInfoUri: ${signavio.platform.url}/p/solman72ConnectorUserinfo
```

```
# maximum level of nested directories to read on an export
maxRecursionLevel: 20
```

Here is how to configure each parameter:

-sharedJwtSecret: This is where you need to set the secret. It MUST be the same in both solman72.yml and configuration.xml. -platform: The URL of the server where you are running Signavio Process Manager. You can name this what you like, but it MUST be the same as the <server> parameter in configuration.xml. -userInfoUri: Uses the same definition as the platform parameter, so this doesn't need to be changed. -username: Username for the database user account. -password: Password for the database user account. -URL: The URL of the database you are using for the Solution Manager connector. Make sure it's different from configuration.xml. -driverClassName: The name of the driver used with your database. This is the same as the driverClass parameter used when setting up Signavio Process Manager. -database encryption secret: When you start the service, you need to configure the user that logs into SAP Solution Manager.

Warning: Once you set the password, do NOT change it again! Otherwise, you won't be able to de-crypt the stored password, and you will no longer be able to use the Solution Manager connector without manual (unsupported) changes to the database.

3.9.2 Configuring the connector

Configuring the connector is similar to the configuration used for Signavio Process Manager. You will use Environment instead of Parameter in the code.

Open context.xml in the Tomcat conf folder, and add the following code snippet:

```
<Environment name="spring.config.location" value="${catalina.base}/conf/solman72.yml" type=
↪"java.lang.String"/>
```

Next, open configuration.xml and add the following:

```
<sharedJwtSecret>SHARED_SECRET_CHANGE_ME</sharedJwtSecret>

<gatewaymappings>solman72=http://localhost:8080/solman72</gatewaymappings>
```

These parameters need to be configured:

-sharedJwtSecret: Just like with the template file, this is where you set the secret. It MUST be the same in both solman72.yml and configuration.xml. -gatewaymappings: This is your host and port. http://localhost:8080 is the default. -solman72 is the name of the directory in the Tomcat webapps folder.

Hint: The most important thing to keep in mind is that parameters within the template and the configuration files need to be kept consistent with each other.

3.9.3 Using Solution Administration

See the Signavio user guide[2] for information on using the connector.

[2] https://docs.signavio.com/userguide/editor/en/workspace_admin/sap_solman/solman72/index.html#solution-administration

3.9.4 Generating the SSL certificate (optional)

This section is only relevant if you are using HTTPS with Signavio Process Manager. See the *Using HTTPS (optional)* (page 38) chapter for additional information.

If you are using HTTPS, then Apache Tomcat requires an SSL certificate with multiple DNS names in order to run the connector.

First, create a configuration file named `selfsigned.conf`, and use that to generate the SSL certificate. Below is a sample configuration file you can use as a template.

```
[ req ]
default_bits = 2048
default_keyfile = localhost.pem
distinguished_name = subject
req_extensions = req_ext
x509_extensions = x509_ext
string_mask = utf8only


[ subject ]
countryName = Country Name (2 letter code)
countryName_default = US

stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = NY

localityName = Locality Name (eg, city)
localityName_default = New York

organizationName = Organization Name (eg, company)
organizationName_default = Example, LLC

commonName = Common Name (e.g. server FQDN or YOUR name)
commonName_default = Example Company

emailAddress = Email Address
emailAddress_default = test@example.com

[ x509_ext ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer


basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
subjectAltName = @alternate_names
nsComment = "OpenSSL Generated Certificate"

[ req_ext ]
subjectKeyIdentifier = hash

basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
subjectAltName = @alternate_names
nsComment = "OpenSSL Generated Certificate"


[ alternate_names ]
DNS.1 = localhost
DNS.2 = 127.0.0.1
DNS.3 = mycompany.com
DNS.4 = signavio.mycompany.com
```

```
# Add these if you need them
# DNS.5 = mycompany.de
# DNS.6 = mycompany.co.uk
# DNS.7 = 127.0.0.1

# IPv6 localhost
# DNS.8 = ::1
# DNS.9 = fe80::1]
```

Next, change the DNS parameters to match the addresses you will use to access the platform. In the command line, use `openssl req -config selfsigned.conf -new -x509 -sha256 -newkey rsa:2048 -nodes -keyout localhost.key.pem -days 365 -out localhost.cert.pem` to generate the certificate. Finally, use the command `openssl pkcs12 -export -out localhost.pfx -inkey localhost.key.pem -in localhost.cert.pem` to generate a PFX file.

3.9.5 Trusting SSL certificates in Tomcat (optional)

The last step is to instruct your Tomcat to trust your self-signed certificate.

Create a trusted Java Keystore. In the command line, add the self-signed certificate to it using the command `keytool -importcert -file localhost.cert.pem -keystore keystore.jks -alias "localhost"`. Choose a new password when prompted.

Configure your Tomcat to trust the keystore that you've just created by starting Tomcat with the Java parameters `-Djavax.net.ssl.keyStore`, `-Djavax.net.ssl.keyStorePassword`, `-Djavax.net.ssl.trustStore`, and `-Djavax.net.ssl.trustStorePassword`. One way of doing it is through the environment variable `CATALINA_OPTS`, like so: .. code:

```
bash_profile
export CLIENT_CERT=/path/to/keystore.jks
export CATALINA_OPTS="$CATALINA_OPTS -Djavax.net.ssl.keyStore=$CLIENT_CERT -Djavax.net.ssl.
↪keyStorePassword=mypassword -Djavax.net.ssl.trustStore=$CLIENT_CERT -Djavax.net.ssl.
↪trustStorePassword=mypassword"
```

Finally, you need to configure your Tomcat to use your newly generate PFX file. Follow the instructions as specified in the *Using HTTPS* (page 38) chapter.

3.9.6 System diagnostics

On-premise customers can use our system diagnostics page to check that their connector installation was successful. See the *Using the Diagnostics Tool to troubleshoot configuration issues* (page 27) chapter for further details.

3.10 Configuring Signavio Process Manager

Before you can start the web application, you need to configure the Signavio Server. The ZIP archive contains the configuration blueprint `configuration.xml`. Place this file together with the Signavio Process Manager license file `configuration_signed.xml` and the `simulation.xml` file into the directory `$TOMCAT_DIR/conf`. `$TOMCAT_DIR` denotes the root directory of your Tomcat instance.

Then, adjust the following properties in the `configuration.xml` file:

- **server**

  The URL of the target server. It must not end with a / (slash).

  Format: `http(s)://<DNS entry or IP address>(:<port>)`

Examples: `http://signavio.mycompany.com`, `http://159.234.37.47:8080`

- **defaultLanguageCode**

  The default language of the system. It is important for systems using Collaboration Hub (guest access). Possible values are:

  - `de` (German)
  - `en` (English)
  - `es` (Spanish)
  - `fr` (French)
  - `ja` (Japanese)
  - `ko` (Korean)
  - `nl` (Dutch)
  - `ru` (Russian)
  - `zh` (Chinese)

- **externalconfigurationpath**

  The absolute path to an existing folder on the target server. The system user who runs the Tomcat Server needs to have write access to this directory. The path must be outside of the `webapps` directory, because during an update of Signavio Process Manager the content of the `webapps` folder is removed. This folder contains files that are cached during run-time of the application. You don't need to backup the content of this folder.

- **Database configuration**

  The template configuration file already contains a default configuration for the connection to a MySQL, Oracle and MS SQL server database. Please use those defaults and configure the variables URL, user name and password.

  - `driverClass`: The Java class name of the JDBC driver.

    Possible values are:

    - **MySQL**: `com.mysql.jdbc.Driver`
    - **Oracle 11, 12**: `oracle.jdbc.driver.OracleDriver`
    - **MS SQL Server**: `com.microsoft.sqlserver.jdbc.SQLServerDriver`

  - `url`: The URL of the database system

    Examples:

    * **MySQL**: `jdbc:mysql://localhost/platform`
    * **MS SQL Server**: `jdbc:sqlserver://localhost;databaseName=signavio;`

  - `username`: The username of the database user account.
  - `password`: The password of the database user account.
  - `dialect`: Hibernate specific SQL dialect.

    Possible values are:

    * **MySQL**: org.hibernate.dialect.MySQL5InnoDBDialect
    * **Oracle**: org.hibernate.dialect.Oracle10gDialect
    * **MS SQL Server**:

      `com.signavio.hibernate.dialect.CustomSQLServerDialect`

- **Email server configuration**:

  Configure the connection to an email sever that will send out system emails like notifications and registration emails.

  - `SMTP_HOST_NAME`: Host name of the email server
  - `SMPT_EMAIL`: Sender mail addresses for outgoing emails
  - `SMTP_AUTH_USER`: Username of the email account to be used
  - `SMTP_AUTH_PWD`: Password of the email account to be used
  - `SMTP_PORT`: Port the email server uses
  - `SMTP_SECURE`: Usage of TLS/SSL. Possible values are `true` and `false`

- `Support email address` (optional):

  This is the receiving address of the support request users can send through the support form out of Signavio Process Manager. If this attribute is empty, the Signavio support address will be used. Please insert the email address of your contact person, for example `<supportMailAddress>support@signavio.com</supportMailAddress>`.

- Maximum diagram count for reports configuration (optional):

  Insert the following line:

  `<maxDiagramCount>500</maxDiagramCount>`

  In this example, the value is increased to a maximum of 500 diagrams. The default value is 250.

In the same directory, add the following lines between the `<Context>` tag to the file `context.xml`:

```
<Parameter name="configfile"
  value="${catalina.base}/conf/configuration.xml" />

<Parameter name="licensefile"
  value="${catalina.base}/conf/configuration_signed.xml" />

<Parameter name="configfile.simulation"
  value="${catalina.base}/conf/simulation.xml" />
```

Note: The location of the configuration files changed with version 9.7.0. Follow *step 6 in the update chapter* (page 48) to move the files to the new location.

Check if the system user that runs the Tomcat has access to newly copied files. If this is not the case, update the corresponding permissions accordingly.

Now, start the Tomcat server.

You can now access Signavio Process Manager in a web browser via the DNS entry or the IP address configured (e.g. `http://signavio.yourcompany.com`).

If you want to change the configuration later, you need to restart the Tomcat server to apply these changes.

Hint: Signavio provides customers of the On Premise Edition automatically with a new license once the old one expires.

3.11 Configuring the GitHub integration for RedHat JBoss BRMS projects (optional)

Note: Configure this integration only if you plan to use the Signavio Process Editor to push diagrams to your RedHat JBoss BRMS projects on GitHub.

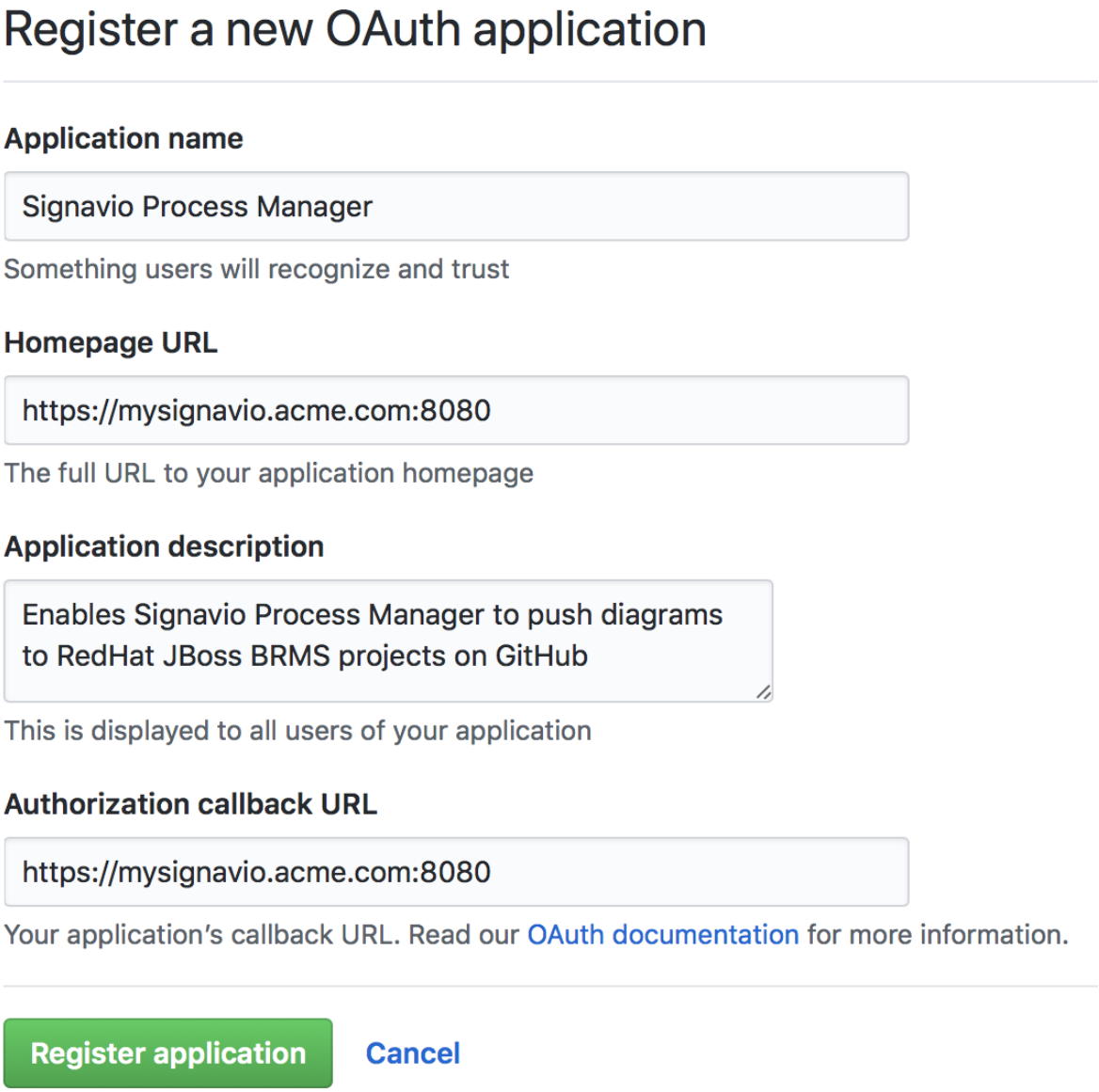
The Signavio Process Editor integration with GitHub[3] allows you to push diagrams from the Signavio Process Editor to your RedHat JBoss BRMS projects for seamless updates.

To configure your system for the GitHub integration, proceed as follows:

1. **Create a GitHub OAuth application.**

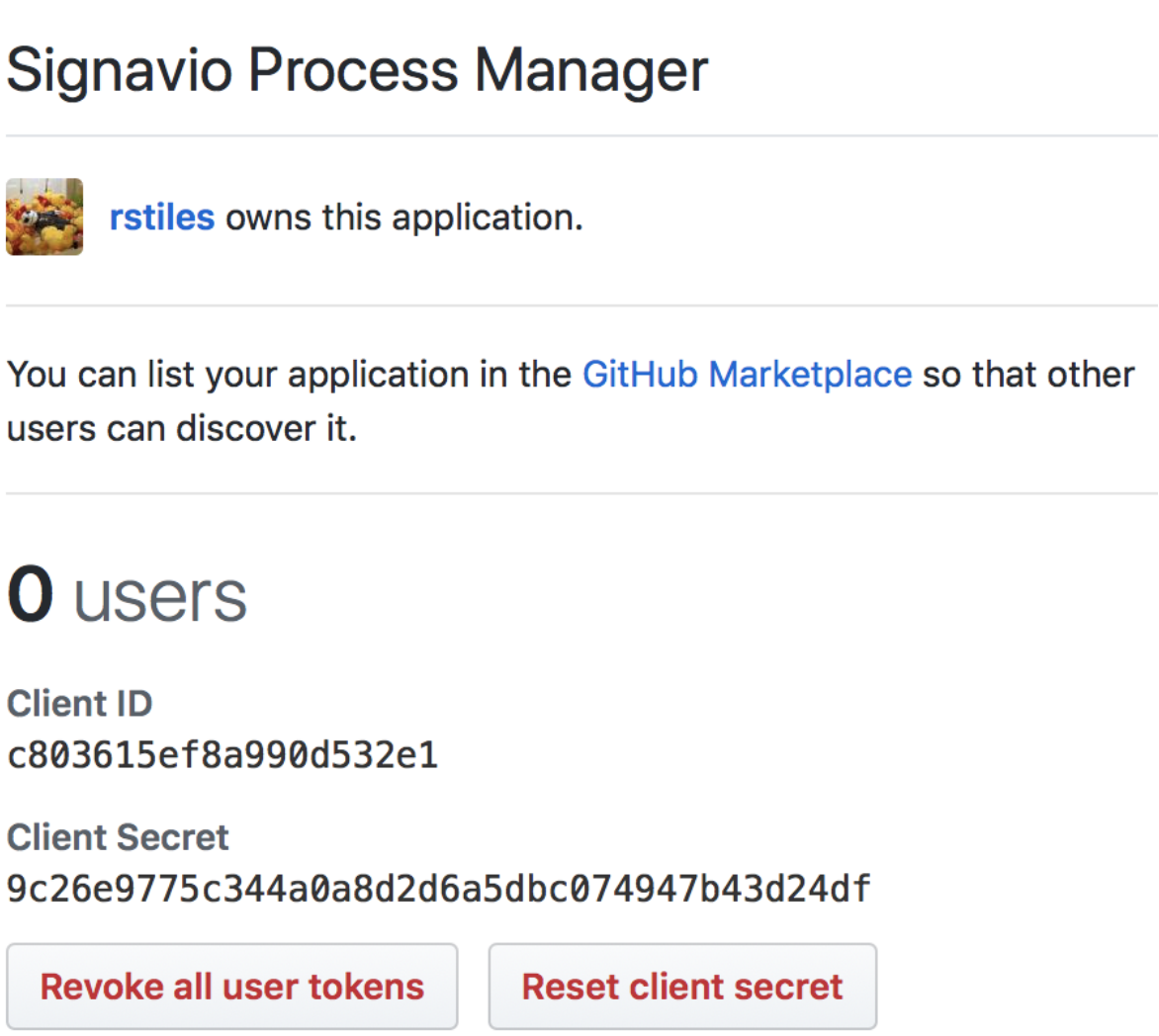
   Log into your GitHub account and open https://github.com/settings/applications/new. Enter an application name, a homepage and a description of your choice. Also enter the base URL of your On Premise as the **Authorization callback URL**, for example `https://mysignavio.acme.com:8080`.

[3] https://editor.signavio.com/userguide/en/explorer/export/redhat_to_github.html



Click **Register application**.

Keep the website open to copy the tokens `Client ID` and `Client Secret` later.



2. **Add the GitHub application credentials to your Signavio Process Editor configuration.**

   Open the file `configuration.xml` in the folder `$TOMCAT_DIR/conf/`, with `$TOMCAT_DIR` being your Tomcat's root directory.

   Add the following lines above the closing `</configuration>` tag.

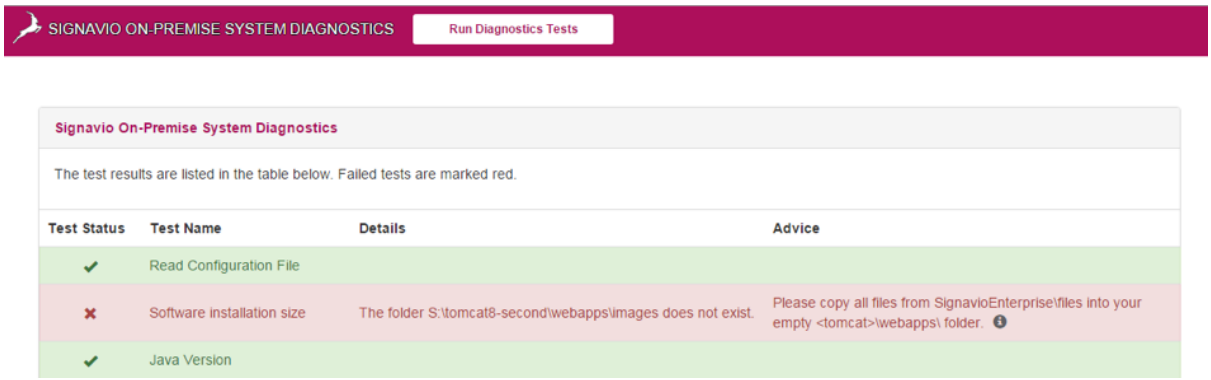
```
<githubClientId><myClientID></githubClientId>
<githubClientSecret><myClientSecret></githubClientSecret>
```

   Replace `<myClientID>` with your *Client ID* and `<myClientSecret>` with your *Client Secret*.

3.12 Using the Diagnostics Tool to troubleshoot configuration issues

To check if everything went fine after starting the Tomcat, visit the URL `<baseURL>/diagnostics` (for example: `https://mycompanysignavio.com:8080/diagnostics`).

Here, you see the following window:



Run Diagnostics Test

Click **Run Diagnostics Test**. The tool is automatically checking the log entries. In case of errors, you receive a problem description and detailed advice on how to proceed.

To prevent users from accessing the diagnostics results, follow the instructions at *Configuring HTTP basic authentication*.

In the code snippets, insert `diagnostics` instead of `register`, respectively

```
<url-pattern>/p/register/*</url-pattern>
```

instead of

```
<url-pattern>/diagnostics/*</url-pattern>.
```

3.13 Registering users

To register a user, open the following URL in your browser:

```
http://<IP address or DNS of the Signavio system>/p/register
```

Fill out the form and click **Register**. Repeat this step for each user you want to register. Users login by entering their email address and a personally chosen password.

Hint: A user's email address is a unique ID. It is not possible to change a user's email address after its registration!

The first registered user is automatically added to the special user group **Administrators**. You can add more users manually to the group. Users who are member of the **Administrators** group operate with extended rights that allow managing user groups or adding modeling guidelines. Therefore, the first user is able to add other users to the administrators group.

3.14 Configuring HTTP basic authentication

To prevent unauthorized users from registering themselves by using the link `http://<IP address or DNS of the Signavio System>/p/register`, you can protect this URL with **HTTP Basic authentication**.

Edit the file `$TOMCAT_DIR/conf/web.xml` and add the following lines to its end before the XML tag `</web-app>`:

```
<security-role>

        <role-name>register</role-name>

</security-role>

<security-constraint>

        <display-name>
                Security constraint for the user registration page
        </display-name>

        <web-resource-collection>

                <web-resource-name>
                        Protected Area
                </web-resource-name>

                <url-pattern>
                        /p/register/*
                </url-pattern>

        </web-resource-collection>

        <auth-constraint>

                <role-name>
                        register
                </role-name>

        </auth-constraint>

</security-constraint>

<login-config>

        <auth-method>
                BASIC
        </auth-method>

        <realm-name>
                Register
        </realm-name>

</login-config>
```

This configuration ensures only Tomcat users with the role `register` can access the resource `/p/register`.

The role `register` needs to be defined and assigned to an existing Tomcat user account.

To do so, edit the file `$TOMCAT_DIR/conf/tomcat-users.xml` and replace its content with the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>

<tomcat-users>

        <role rolename="register"/>

        <user username="signavio" password="signavio" roles="register"/>

</tomcat-users>
```

These lines define the role `register` and create a user with the username `signavio`, the password `signavio` and the role `register`.

Please customize the attribute values for `username` and `password` and share this information only with authorized persons who are allowed to register new users in the Signavio system.

3.15 Configuring Collaboration Hub & Active Directory (optional)

If you have purchased Collaboration Hub and want to use single-sign-on, the you need to configure the connection to your Active Directory service.

Hint: If you use Kerberos-SSO functionality, you need to install the Signavio server on a Microsoft Windows 64 bit operating system of a machine that is within your Active Directory domain.

Hint: If you want to test access to your Active Directory, you may use the **LDAP Browser** by Softerra. For example, you can check, if the user name and password is correct and if the user can access relevant parts of the Active Directory. You can download the Softerra LDAP Browser free of charge at:

http://www.ldapbrowser.com/download.htm

Hint: Before activating or deactivating the SSO-functionality you should unpublish all diagrams. Otherwise diagrams may not be correctly published for the some users.

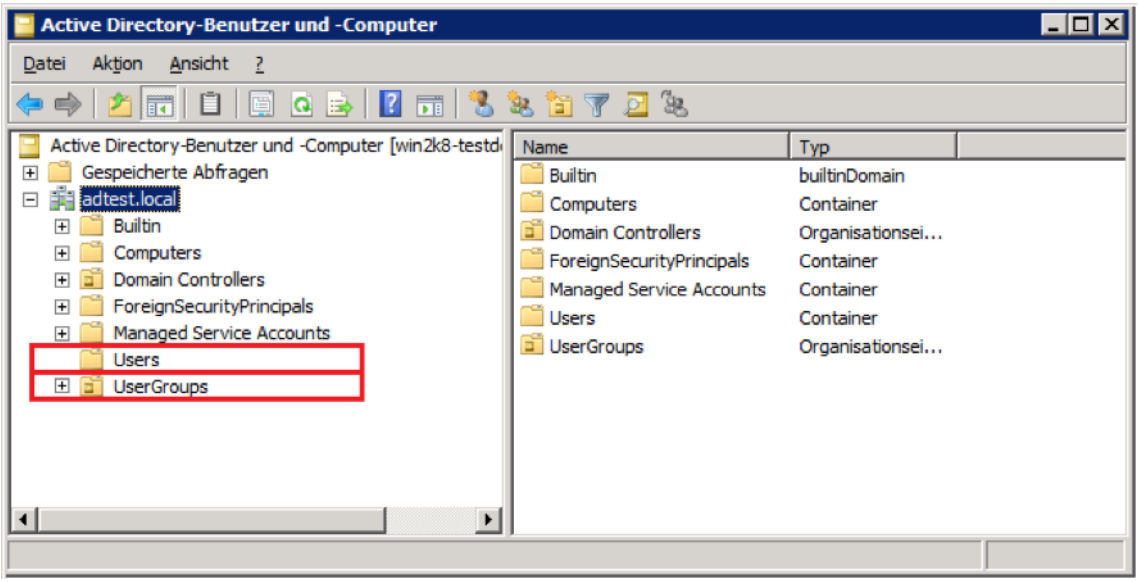
The configuration template `configuration.xml` contains an example configuration of an Active Directory connection:

```
<ldap>

  <ldapHost>ldap://192.168.0.100/</ldapHost>

  <ldapSearchRoots>

    <ldapSearchRoot>CN=Users,DC=company,DC=com</ldapSearchRoot>

    <ldapSearchRoot>CN=Groups,DC=company,DC=com</ldapSearchRoot>

  </ldapSearchRoots>

  <ldapUser>CN=LdapUser,CN=Users,DC=company,DC=com</ldapUser>

  <ldapPw>password</ldapPw>

  <ldapSsoMode>KERBEROS</ldapSsoMode>

  <ldapModelerSsoSupported>true</ldapModelerSsoSupported>

  <ldapAdminMail>admin@company.com</ldapAdminMail>

</ldap>
```

For requests to the Active Directory, LDAP interfaces are used. To enable the connection, adjust the `configuration.xml` file as follows:

- **ldapHost**: URL for the access to the Active Directory using LDAP.

- **ldapSearchRoots**: Multiple folder objects that will be used as root directory for the search. Specify these objects as **Distinguished Names**. Search roots can be either **Organizational Units** or **Containers**. Note user groups are **not** containers for users, but references to those. Below you see an example of a structure of an Active Directory in the domain `adtest.local`:
  - If the containers *Users* or user *UserGroups* are in the domain's root node, all folders highlighted in red have to be marked as `searchRoots`:

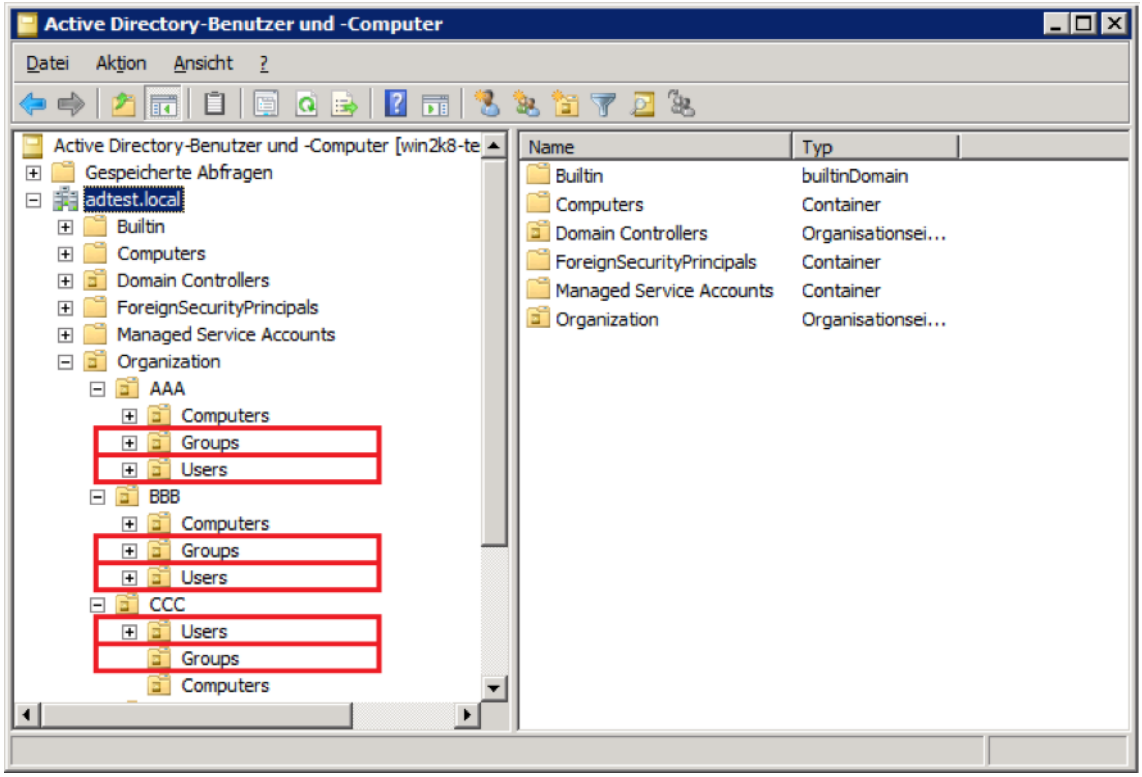


Example: user groups on root level in an Active Directory

The *Distinguished Names* in this case are:

```
CN=Users,DC=adtest,DC=local
OU=UserGroups,DC=adtest,DC=local
```

If the Active Directory is divided by location or organization units, the sub folders containing groups and user have to be marked as `searchRoots`:



Example: user groups on organization level in an Active Directory

The Distinguished Names in this case are:

```
OU=Groups,OU=AAA,OU=Organization,DC=adtest,DC=local

OU=Users,OU=AAA,OU=Organization,DC=adtest,DC=local

OU=Groups,OU=BBB,OU=Organization,DC=adtest,DC=local

OU=Users,OU=BBB,OU=Organization,DC=adtest,DC=local

OU=Groups,OU=CCC,OU=Organization,DC=adtest,DC=local

OU=Users,OU=CCC,OU=Organization,DC=adtest,DC=local
```

- **ldapUser**:

  User object for the access to the Active Directory. Depending on the version and the configuration of the active directory, there are three different notations for the user name:

  Please try the different notations in this order. If it does not work try the next notation:

  1. Distinguished Name (e.g.: `CN=LdapUser,CN=Users,DC=company,DC=com`)
  2. User UPN logon name (e.g.: `LdapUser@company.com`)
  3. SamAccountName (e.g.: `company\LdapUser`)

- **ldapPw**:

  Password of the user object.

- **ldapSsoMode**:

  Used mechanism for the user Single-Sign-On. Might be `KERBEROS` or `LDAPQUERY` (authentication using an LDAP user name and password). If you use Kerberos, Signavio Process Manager must be installed on a Microsoft Windows 64 bit operating system. If you use LDAP-query based authentication, you can also specify an `ldapQueryLoginPattern`.

- **ldapQueryLoginPattern** (optional): Allows you to specify a completion pattern for the user name provided to the LDAP-query based authentication.

  Example: Using the pattern `$login$@company\.com`, you can configure that instead of the user input `j.doe` the following user name is actually used for authentication: `j.doe@company\.com`.

- **ldapModelerSsoSupported**

  Controls, whether Active-Directory users can login to Signavio Process Manager as modeling users without providing a separate Signavio password. Possible values are true and false (default).

  To use this SSO for a specific Active-Directory user, a modeling user must be registered within Signavio Process Manager that has the same email-address as the Active-Directory user.

- **ldapAdminMail**

  Email address of the administrator who is responsible for the LDAP configuration. This is used for system notifications and thus has to be defined.

Copy the customized `configuration.xml` into the directory `$TOMCAT_DIR/conf/` and restart the Tomcat server.

Now, grant read access to the BPM Collaboration Hub to users and/or user groups according to the chapter Granting Collaboration Hub read access to Active Directory service users and user groups[4] of the Signavio user guide.

With Collaboration Hub, you purchase a contingent of a specific number of Active Directory users who are allowed to get read access to models that were published in Signavio Collaboration Hub. When calculating the number of active users, the system considers all users that have access to any published diagram.

If the quantity of Collaboration Hub users reaches 80% of the contingent, the system will automatically inform your workspace administrator by email.

If the contingent is exceeded, the read access to the Signavio process portal is temporarily deactivated. However, licensed users will still be able to log in.

Hint: To activate the automatic authentication for the Internet Explorer using a Kerberos user, add the host name of the Signavio server to the security zone **Local intranet** and ensure the option **user authentication > registration** is set to **Automatic logon only in Intranet zone**.

To activate Kerberos authentication using Mozilla Firefox, go to **about:config** via the address bar of the browser. There, set the variable `network.negotiate-auth.trusted-uris` to the Signavio server URL as defined in the `configuration.xml` file. If the variable is already set, extend it as a comma-separated list with the server URL.

3.16 Configuring Signavio Collaboration Hub SharePoint component (optional)

This section describes the installation process of the optional **Signavio Collaboration Hub Web Part**. With the Signavio Web Part, you can embed Collaboration Hub into a Microsoft SharePoint system.

You need a Microsoft Office SharePoint Server (http://sharepoint.microsoft.com) for the installation

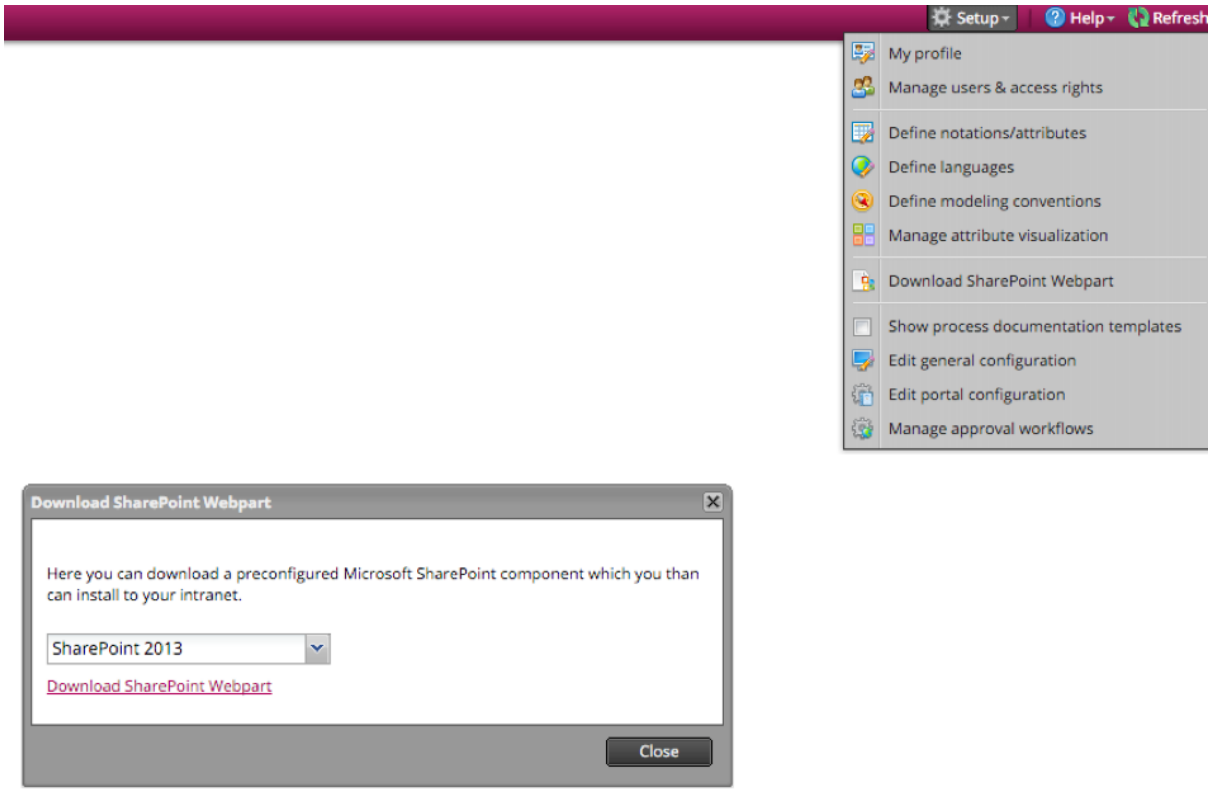
Note: Integrating **SharePoint for Office 365** (SharePoint Online) is only possible with our software-as-a-service offer.

[4] https://editor.signavio.com/userguide/en/html/workspace_admin/Granting-Collaboration-Portal-read-access-to-directory-service-users-and-use
html

Important: Since Microsoft® stopped supporting Microsoft® SharePoint® 2007 and 2010 on October 9, 2012[5], respectively on October 13, 2015[6], it is getting increasingly difficult to provide Collaboration Hub for these SharePoint versions. As we want to keep Collaboration Hub up to date and to provide the best features possible, we dropped support for Microsoft® SharePoint® 2007 and 2010 on **April 1, 2017**.

The Web Part allows a preconfigured view of Collaboration Hub. The Web Part needs to be purchased as a separate component.

Download the Web Part from the Signavio Explorer at **Setup** - **Download SharePoint Web Part**:



Download the Signavio SharePoint Web Part

Copy the file `SignavioViewer2013_2016.wsp` into a directory of your choice on the SharePoint server. Install the SharePoint Solution as described in the corresponding SharePoint manual.

3.16.1 Installing the Web Part

The following section explains how to install the Web Part with different Microsoft SharePoint Server versions. In our examples, the parameter `<SharePointServerURL>` represents the SharePoint server's web application URL.

The Signavio Hub Web Part uses an SSL connection to communicate with the Signavio server. In case you configured very strict security policies for your SharePoint server, you might need to import the SSL certificate authority used by Signavio into Microsoft SharePoint's **Trusted Root Certificate Store**. The certificate file (`geotrust.cer`) can be downloaded at https://editor.signavio.com/sharepoint/geotrust.zip (Europe), https://app-us.signavio.com/sharepoint/geotrust.zip (US) or https://app-au.signavio.com/sharepoint/geotrust.zip (Australia/APAC).

Follow these steps to install the geotrust certificate:

[5] https://support.microsoft.com/en-us/lifecycle?p1=11373
[6] https://support.microsoft.com/en-us/lifecycle?p1=11373

- Ensure the Windows service **SharePoint Administration** runs.
- Open the **Microsoft SharePoint 2013/2016 Central Administration**.
- Navigate to **Security** > **Manage trust settings** and click **New**.
- Enter a name, for example *Signavio*, and browse to the `geotrust.cer` file.

To install the Web Part, please open the management shell of the SharePoint server and execute the following commands:

```
Add-SPSolution -LiteralPath <Path>\SignavioViewer2013_2016.wsp

Install-SPSolution -Identity SignavioViewer2013_2016.wsp -WebApplication
<SharePointServerURL> -GACDeployment
```

You can check the status of the installation through the following `http://<SharePointServerURL>:<ADMIN_PORT>/_admin/SolutionStatus.aspx?ItemName=signavioviewer2013_201`

You find further information about installing a .wsp file on a Microsoft SharePoint Server at https://technet.microsoft.com/en-us/library/cc262995.aspx?f=255&mspperror=-2147217396

3.16.2 Configuring the Web Part

After installing the SharePoint Solution on a SharePoint Server, you can include the Web Part in the desired SharePoint page(s).

1. Navigate to **Page** and click **Edit page**.
2. Now, click **Add a Web Part**.
3. Select **Signavio Collaboration Hub Web Part** and add it.
4. Use **Edit** to add the **Signavio URL** to the section **Signavio** on the right. It either directs to the Signavio server or defines an URL to a specific start diagram (as an URL from Collaboration Hub preview). Additionally, you can customize the look and feel of the Signavio component, for example by configuring the title or size.

You find more information about embedding a Web Part at: http://office.microsoft.com/de/HA010097463.aspx.

3.17 Configuring approval workflows with Signavio Workflow Accelerator (optional)

Signavio provides Signavio Workflow Accelerator to customers who purchased a corresponding subscription. If your organization wants to use the **approval workflows** feature, you need to install Signavio Workflow Accelerator on your server.

Please refer to the Signavio Workflow Accelerator administrators guide at https://docs.signavio.com/adminguide/workflow/en/ for more information about the system requirements and the installation process.

Hint: Before installing Signavio Workflow Accelerator, please ensure your Signavio version is compatible with the Signavio Workflow Accelerator version you plan to install. You find a corresponding compatibility list at https://docs.signavio.com/adminguide/workflow/en/signavio.html#compatibility-list.

3.18 Configuring the Signavio Simulation Server (optional)

The execution of process simulations is performed in a standalone web application that is usually deployed on the same Tomcat server as the rest of the application. To avoid an overload of the web server by a complex simulation scenario, a simulation instance will be canceled when reaching a preconfigured threshold. You can increase these thresholds on your own risk. To avoid an overload of the web server, you can also deploy the simulation component on a separate Tomcat server.

3.18.1 Adjusting properties of the Signavio Simulation Server

For this you need to adjust the file `simulation.xml` you find at `$TOMCAT_DIR/conf/`.

Uncomment the respective XML-tags and fill the tags as follows:

- `maxResults`: Defines the maximum number of simulation results that will be cached in the Simulation Server at a time. The default value is 100.
- `maxCachedInstances`: Defines the maximum number of running simulation instances that will be cached in the Simulation Server at a time. The default value is 100.

$$\text{NrOfParallelNCaseRuns} \times \frac{maxLogEntries \times 15}{100.000}\text{MB}$$

- `resultTimeout`:

  Defines the amount of time (seconds) until a simulation result is considered as old and can be deleted. The default value is 600.

- `instanceTimeout`: Defines the amount of time (seconds) until a simulation instance is considered as old and can be deleted. The default value is 600.

Hint: If you want to run the simulation server with the defaults, you don't need to edit the values after uncommenting them.

3.18.2 Installing Signavio Simulation Server in a separate Tomcat instance (optional)

Important: In most cases it is not necessary to install the Signavio Simulation Server on a separate Tomcat. The only good reason for doing so is to prevent performance problems if both of the following conditions are valid:

- You have purchased the Enterprise Edition.
- Many of your business process modelers are frequently using the n-case simulation feature.

In order to run the Simulation server on a separate Tomcat, install the Tomcat and configure it just as the Signavio Tomcat (see: *Configuring the web server* (page 12)).

Hint: In case the second Tomcat runs on the same server/VM, please make sure it is running on a different port than the Signavio Tomcat.

Hint: Make sure to always update Signavio Process Manager together with the Simulation Server. Different versions are incompatible with each other.

Now, proceed as follows:

1. Stop the Tomcats (if running).
2. **Move** (not copy!) the folder `simulationservice` from the webapps-directory of the Signavio Tomcat to the `webapps` directory of the Simulation Server Tomcat.
3. Adjust the file `simulation.xml` you find at `<signavio-tomcat>\conf`. Uncomment the corresponding XML-tags and fill the tags as follows:
   - `host`: Defines the host name of the server where the Simulation Server is deployed, for example `http://mysignaviosimulationservice:8180`.
   - `path`: Defines the URL path at which the Simulation Server is deployed, normally `/simulationservice/`.
4. Start both Tomcats (the order does not matter).

3.19 Configuring the Apache Solr server (optional)

To search diagrams, folders, uploaded documents and dictionary items the Signavio application server employs Apache Solr. Usually, the Solr server is deployed on the same Tomcat as the rest of the application.

However, you can deploy the Solr server on a separate web server. If your Signavio system fulfills one the following conditions, it may be useful to employ a separate server for better performance:

- You are using the system with more than 1,000 modelling users.
- You are using the system with more than 50,000 Collaboration Hub users.
- Your process landscape contains more than 5,000 diagrams.
- You are running Signavio on a web server cluster.

To run the Solr server on a separate Tomcat, install the Tomcat and configure it just as the Signavio Tomcat (see: *Configuring the web server* (page 12)).

Hint: In case the second Tomcat runs on the same server/VM, please make sure it it runs on a different port than the Signavio Tomcat.

Afterwards, proceed as follows:

1. Stop the Tomcats (if running).
2. **Move** (not copy!) the folder solr from the webapps-directory of the Signavio Tomcat to the webapps-directory of the Solr Tomcat.
3. Adjust the file `configuration.xml`, located at `<signavio-tomcat>/conf/`. Add the following line at the end of the file, but before the line `</configuration>`:

   ```
   <solrUrl>http://localhost:8080/solr</solrUrl>
   ```

   Replace the URL `http://localhost:8080/solr` with the URL of the Solr server. Please ensure you also define the context path of the web application (default is `solr`).
4. Start the Solr Tomcat.
5. Start the Signavio Tomcat.

Hint: Before you start the Signavio server, always ensure the Solr server is running. Otherwise, system cannot create the search index.

3.20 Configure image server (optional)

The configuration procedure of the image server is similar to the configuration of the Solr server.

Follow the instructions at *Configuring the Apache Solr server (optional)* (page 37).

This time, move the folder `imageservice` from the `webapps` directory of the Signavio Tomcat to the `webapps` directory of the image service Tomcat and add the following line to the `configuration.xml` file:

```
<imageServiceHost>http://localhost:8080</imageServiceHost>
```

Replace the URL `http://localhost:8080` with the URL of your image server.

3.21 Configuring the mailing server (optional)

The mailing service that is used for sending emails is usually deployed on the same Tomcat as the rest of the application. However, you can deploy the mailing service on a separate server. If you are running a Signavio system that fulfills one the following conditions, it may be useful to use a separate server for better performance:

- You are using the system with more than 1,000 modeling users.
- You are using the system with more than 50,000 portal users.
- You are running Signavio on a web server cluster.

In order to run the mailing server on a separate Tomcat, install the Tomcat and configure it just as the Signavio Tomcat (see: *Configuring the web server* (page 12)). Additionally, ensure that the mailing server can access the Signavio database. We recommend to use the same database user as for the Signavio server.

Hint: In case the second Tomcat runs on the same server/VM, please make sure it is running on a different port than the Signavio Tomcat.

Afterwards, proceed as follows:

1. If running, stop the Tomcats.
2. **Move** (not copy!) the folder `mailingservice` from the `webapps` directory of the Signavio Tomcat to the `webapps` directory of the mailing service Tomcat.
3. Adjust the file `configuration.xml` you find at `<signavio-tomcat>/conf`.
4. Start the Signavio Tomcat.
5. Start the mailing server Tomcat.

Hint: Always ensure the Signavio server is running before the mailing server is started. Otherwise, the initialization of the mailing service fails.

3.22 Using HTTPS (optional)

Signavio Process Manager supports encrypted communication via HTTPS. We recommend using HTTPS for example, if the users access the Signavio system via an insecure network connection.

If you want to use the HTTPS protocol, some further configurations in the software are necessary. It is required that the respective administrator has some knowledge in the usage of HTTPS in combination

with the Apache Tomcat. For further information and help regarding the installation and configuration please consult the documentation of the Apache Tomcat.

Hint: Please make sure that the certificate is issued by a trusted Certificate Authority (CA) and please make sure that the JVM is trusting that certificate and CA. Therefore, both certificates must be added to the Java Truststore.

An incorrectly configured HTTPS connection usually typically causes the following problems:

- The search of diagrams and dictionary entries returns no results.
- The Dictionary is empty.
- Diagrams can't be exported as a PNG, the diagram preview in the explorer shows no picture of the diagram, the BPM Collaboration Portal shows no graphical representation and in the Simulation and Comparator no diagram is displayed.
- The Simulation can't be executed.
- The system doesn't send notification emails.

The reason for these effects is that the different features are provided by different web services that communicate via HTTP(S).

To ensure a stable connection, the JVM instance must accept the used certificate. If you observe one or more of these effects, please check the Tomcat log files. Usually, the error messages contain helpful information about the problem.

3.22.1 Tomcat SSL Certification

For Tomcat 8.0, you will find detailed instructions for SSL certification in the official Apache Tomcat documentation at:

https://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html

Please make sure to set UTF-8 as the URIEncoding of the HTTPS connector.

A readily prepared connector configuration could look like this:

```
<Connector port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="150"
  SSLEnabled="true" scheme="https" secure="true"
  keystoreFile="Path/to/.keystore"
  keystorePass="mypassword"
  clientAuth="false" sslProtocol="TLS"
URIEncoding="UTF-8" />
```

For Tomcat 8.5, you will find detailed instructions for SSL certification in the official Apache Tomcat documentation at:

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

Please make sure to set UTF-8 as the URIEncoding of the HTTPS connector.

Note: For Tomcat 8.5, the SSL protocol has changed. The old TLS protocol is no longer supported by all browsers. As a result, you will not be able to reach your application if you try to use the default TSL protocol. To fix this, you need to define an additional TLS version.

Your connector could look like this:

```
<Connector port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="200"
  SSLEnabled="true" scheme="https" secure="true"
  keystoreFile="D:/cert/localhost.pfx"
  keystorePass="123456"
  clientAuth="false" sslProtocol="TLSv1+TLSv1.1+TLSv1.2"
  keystoreType=''PKCS12''
  URIEncoding="UTF-8" />
  compressableMimeType=''text/css,text/plain.image/svg+xml,application/xhtml+xml,text/html,text/
↪xml,text/javascript,application/xml.application/x-javascript,application/javascript''
  compression=''on''
  compressionMinSize=''2048''
  maxHttpHeaderSize=''16384''/>
```

For really large active directories, you will need to double or quadruple the `maxHttpHeadersize` parameter. Otherwise, Tomcat 8.5 will time out.

Hint: The `keystore` lines are optional, and can be changed depending on what kind of certificate you want to use (our example is for PFX). Edit your code accordingly.

At this time, **Signavio does not support Tomcat 9.0**.

3.22.2 Using an Apache web server in front of the Tomcat to enable SSL (alternative to 3.20.1)

Instead of configuring the Tomcat to use SSL, you can setup an Apache web server (that uses SSL) and have it handle the external traffic for the Tomcat.

The configuration procedure is described in the official Apache Tomcat documentation at https://tomcat.apache.org/connectors-doc/webserver_howto/apache.html

3.22.3 Signavio system configuration

In addition to the regular installation procedures, please also follow these steps.

The Tomcat server must be stopped for this.

Important: These instruction assume that the Signavio Application Server, the Mailing Server, the Image Server, the Simulation Server and Apache Solr run on the same machine. If this is not the case, adjust the corresponding URLs according to your system setup and **don't** restrict the Tomcat Connector in step 1) to listen to localhost only (no `address=''127.0.0.1''`). Instead, you can restrict access with a Remote Address Filter[7] , for example.

To configure the Signavio system to work properly with HTTPS, proceed as follows:

1. Activate an **HTTP connector** in the `server.xml` file (in the `conf` directory of your tomcat) as described at **Configure Apache Tomcat.** Make sure to have the connector listening to local requests only (`address=''127.0.0.1''`). A readily prepared connector configuration typically looks like this:

   ```
   <Connector
     address=''127.0.0.1''
     port="80" protocol="HTTP/1.1"
   ```

[7] http://tomcat.apache.org/tomcat-8.0-doc/config/valve.html#Remote_Address_Filter

```
  connectionTimeout="20000"
  redirectPort="8443"
  URIEncoding="UTF-8"
  compression="on"
  compressionMinSize="2048"
/>
```

Important: It is important that all other services of the application make requests against this connector (see below). If they are configured to use the HTTPS connector instead, most probably some features will not be reliably available.

1. Update the Signavio configuration file `configuration.xml` in the folder `$TOMCAT_DIR/conf/`:

   (a) Add the following lines at the end of the file but before the end tag `</configuration>`:

```
<solrUrl>
  http://localhost/solr
</solrUrl>
<imageServiceHost>
  http://localhost
</imageServiceHost>
<platformHostForMailingService>
  http://localhost
</platformHostForMailingService>

<decisionTableVerificationHost>
  http://localhost
</decisionTableVerificationHost>

<bdmSimulationServiceHost>
  http://localhost
</bdmSimulationServiceHost>

<droolsExportServiceHost>
  http://localhost
</droolsExportServiceHost>
```

   (a) In the same directory, in the file `simulation.xml`, make sure the host entry looks like this:

   ```
   <entry key="host">http://localhost</entry>
   ```

2. Update the Apache Solr server:

   (a) Open the file `$TOMCAT_DIR/webapps/solr/META-INF/context.xml`

   (b) Add your IP with a leading `|` to the attribute `allow` of the xml element `Valve`.

   For example, if the IP of the server is `192.168.80.152`, the entry should look like this:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
  allow="localhost|127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|
  192.168.90.152"
/>
```

3. Ensure your configuration is correct. Proceed as described at *Using the Diagnostics Tool to troubleshoot configuration issues* (page 27).

3.23 Installing Signavio Process Manager on a Tomcat cluster (optional)

You can deploy Signavio Process Manager on an Apache Tomcat cluster, with all nodes using the same database.

Using a Tomcat cluster improves the **performance** (scale on high load) and **availability** of your system.

However, we recommend you to **not** deploy Signavio Process Manager in a cluster, to avoid unnecessary administrative overhead.

A system instance that runs on one Tomcat scales well even with more than 1000 modeling users, as long as you increase the memory assigned to the Tomcat (for example to 16GB).

To increase the availability, install a passive web server or to install the web server as a virtual machine on a high-available infrastructure. This increases the availability on server level and not on application level. Furthermore, you can monitor the Tomcat server so that it can quickly be restarted - automatically or manually - if performance issues occur.

If you want to deploy Signavio Process Manager on a Tomcat cluster, please consider the following:

- You need an additional Signavio license for each Signavio server.
- Ensure each Tomcat instance uses a unique JVM-Route. Edit the Tomcat configuration file `$TOMCAT_DIR\conf\server.xml` and add the attribute jvmRoute to the XML Element Engine, for example:

  ```
  <Engine name="Catalina" defaultHost="localhost" jvmRoute="UNIQUE_ID">
  ```

- Use a tomcat session cluster.
- Configure the load balancer to forward all requests of the same session to the same application server (*sticky sessions*).
- Ensure the mailing server is only deployed on one Tomcat instance (see: 3.17 Configure Apache Solr server (optional)). Otherwise, emails will be send multiple times (once for each Tomcat instance).
- Ensure that the Solr server is only deployed on one Tomcat instance (see: 3.17 Configure Apache Solr server (optional)).
- For each Tomcat instance, adjust the file `configuration.xml` located at `<signavio-tomcat>/conf/` as follows:
  - Add the following line:

    ` <enableMultiplePlatformInstances>true</enableMultiplePlatformInstances> `
  - Set the property `enableInternalKeyValueCache` to `false`. If the entry doesn't exist, add the following line before the closing `</configuration>` tag:

    ` <enableInternalKeyValueCache>false</enableInternalKeyValueCache> `

3.24 Conducting manual functional tests

To ensure that your system is operating properly after installation or update, conduct the following manual tests:

- (Only new/first installation): Register at least two Signavio users.
- Log in as a modeling user.
- Create a folder and a diagram (incl. dictionary entries).
- Publish the diagram in Collaboration Hub.
- Send an invitation to comment on a diagram.

- Create a process documentation handbook.
- Create an Excel report.
- Use the BPMN Simulation.
- Use the Dictionary (for example, edit an existing entry).
- Log out as a modeling user.
- Open of Collaboration Hub as a portal user:
  - (Only if configured): Is the user correctly recognized in the Single Sign-On?
  - Is the diagram you published before available?
- Check if notification mails are sent with the configured frequency.

Chapter 4

Operating guide

This chapter guides you through operating Signavio Process Manager.

4.1 Accessing Signavio Process Manager as a user

You access Signavio Process Manager with any modern web browser (see https://www.signavio.com/browser-compatibility/ for a list of supported browsers).

In case you have problems connecting to Signavio Process Manager, make sure your browser's network and proxy settings are correct.

To login to the Signavio web application you need to have a registered user (and their email address and password).

Collaboration Hub without single-sign-on allows any user who can access the Signavio server's URL to view published diagrams in your browser. If some users within your network shouldn't have access to Collaboration Hub, you have to manage these access restrictions on network level.

In contrast, Collaboration Hub with single-sign-on allows you to define user and group specific access rights for each diagram based on your Active Directory service.

4.2 Backing up your data

The database contains all relevant data for a secure recovery of all your repository information.

The only items that are not stored in the database are:

- the search index (that will be generated automatically when you restart the Tomcat server,
- user sessions (that means users have to login again after you restart the Tomcat sever).

Based on your database system and existing infrastructure, a database backup may rely on existing backup strategies. We recommend you to execute at least one backup per day and implement a rotating backup strategy with monthly backups.

You should store the backups on different hardware at another physical location or on external hardware, to protect the backups - for example - from fire.

4.3 Typical malfunctions

This section provides an overview over the most frequent causes for system malfunctions and their possible solutions:

The Tomcat server or the database don't start up after maintenance activities. Try to login each time after maintenance activities to check whether the system started properly.

Not enough disk space is available for the database. Each diagram version requires approximately 1 MB disc space. If a diagram is saved ten times, 10 MB of data are used. We recommend calculating enough disk space and monitoring it on operating system level. For example, a note could be sent to the administrator if the system is running out of disc space.

There was not enough memory allocated to the Tomcat server. This manifests in frequent swapping and slow diagram accesses – even `OutOfMemoryExceptions` and a crash of the Tomcat server may occur. Allocate more memory to the Tomcat server. See chapter "3.5.2 Configure Apache Tomcat" (page 9).

Your browser redirects you to localhost when you enter the Signavio server URL into the web browser. You didn't configure the server correctly. Please ensure that the installation instructions have been followed correctly and restart the Apache Tomcat.

Umlauts are not displayed as expected in Signavio Process Manager UI and exported documents or pictures (PN The Apache Tomcat is not using UTF-8 encoding. Please check the operating system, database and web server configuration (see installation instructions).

The system doesn't send notification emails. Re-check the server's SMTP configuration (see installation instructions).

Users within your company/network receive notification emails, external users don't. Some relay configuration of your email server prevents it from sending emails to external users. For most email servers allowing anonymous access (if no username and password was set during configuration of the email server integration) this is the default configuration. Contact the administrator who is responsible for your email server.

The Tomcat server doesn't start up properly and writes 'Waiting for changelog lock...' into the log file. If the Tomcat startup was canceled previously (for example due to a lack of memory), this error may occur. Stop the server before fixing the defect. Afterwards, log into the database and open the table `DatabaseChangelogLock`. It usually contains just one line. Set the value of the variable `LOCKED` from `1` to `0`. Start the Tomcat server again.

It is not possible to save very large diagrams With the default Tomcat settings, it is not possible to save very large diagrams (> 2 MB of mere textual representation). You can increase the file size limit by adding the `maxPostSize` property to connector definitions in the Tomcat's `server.xml` file. Read more at *Configuring the web server* (page 12).

Some features don't work as expected.

- The search of diagrams and dictionary entries returns no results.
- The Dictionary is empty.
- Diagrams can't be exported as a PNGs, the diagram preview in the explorer shows no picture of the diagram, the BPM Collaboration Hub shows no graphical representation and in the BPMN Simulation and Diagram Comparison no diagrams are displyed.
- The BPMN Simulation can't be executed.
- No daily notifications are sent out.

The different features are provided by different web services which communicate via HTTP(S). For a smooth communication it is necessary that the JVM instance accepts the used certificate. If you observe one or more of these effects please check the Tomcat log files. Probably, the error messages contain helpful hints on what the problem is.

4.4 Maintaining the server

When you need to restart the system - for example because of an update - consider the following issues.

4.4.1 Inform your users

If maintenance activities require a restart of the system, inform all users that the server is temporarily unavailable.

As a workspace administrator of Signavio Process Manager you can access a list of all users to inform them about updates and possible downtimes.

4.4.2 Managing the database during a restart

Signavio Process Manager works exclusively on database systems that support transactions.

Therefore, a **restart** of the database doesn't impact data consistency. However, you must not abort the **initialization** of the server, because it includes operations not all database system run in transactions. Therefore, you **must not** shut down the system during startup. In case you need to shut down the database system, **always shut down the Tomcat server before**.

4.4.3 Managing user sessions during a restart

Every user session will be kept at a restart of the Tomcat server. Therefore, users who are currently logged in stay logged in. Upon shut down, the user sessions data is stored in a file on the Tomcat server (by default `$TOMCAT_DIR/work/Catalina/localhost/\_/SESSIONS.ser`). The system reads the file again during startup and restores the sessions. If the system interrupt only lasts a short time, the modelers who are currently logged in may not notice the restart at all. When users try to save a diagram while the Tomcat server is down, they receive an error message. As soon as the server is restarted, the save process can be repeated and will succeed.

4.5 Monitoring

To avoid unnecessary downtime, you should continuously monitor both the database server and the Tomcat server. For the database server it is most important to monitor the used disc storage. You may set up an observation routine that checks if the server runs out of disk space or if there are problems with creating backups. If the database server responds slowly, you should re-check its configuration (for example, if the database caches are configured correctly or if the data needs to be read from disk each time a request occurs).

You can check the plain availability of the Apache Tomcat server by sending an `http` request and logging into the Signavio system. Additionally, you should monitor the Tomcat server's process and its memory usage. If the process has to touch the swap-section and needs to outsource memory pages to the disc, the system responds very slowly. In such a case, make further memory available.

4.6 Updating the database or operating system

The Signavio application employs JDBC and Apache Tomcat to abstract from the database system, respectively the operating system.

Generally, an update of your database or operating system can be conducted without consulting the Signavio Support Team.

However, you need to ensure the target version of the database/operating system is supported.

Important: Always backup your data before you update. On shut down, stop the Tomcat server before stopping the database system.

4.7 Moving the server to new hardware

Moving the server to a different hardware is no problem at all. Copy your license file and create a dump of your database. Now, install the system from scratch on your new machine and import the database dump there. Ensure all external systems are accessible from the new server (email server, database, Active Directory, SharePoint etc.).

Important: Always backup your data before you migrate. On shut down, stop the Tomcat server before stopping the database system.

Chapter 5

Updating Signavio Process Manager

Important:

- If you use a MS SQL database and update from Signavio Process Manager version 10.x or lower to version 11.x or higher, download and install the new JDBC driver as described at *Download JDBC driver for Microsoft SQL Server* (page 18). **Delete the old file** `sqljdbc4.jar` you find in the `lib` folder of the Apache Tomcat directory.
- If you use an Oracle database and update from version 9.x to version 10.x or higher, download the new Oracle JDBC driver as described at *Download JDBC driver for Oracle* (page 18) and replace the old driver with the downloaded files.
- If you are using Signavio Process Manager with approval workflows, please ensure the to-be-installed version of Signavio Process Manager is compatible with the Signavio Workflow Accelerator version you are using. You find a corresponding compatibility list here in the Workflow Accelerator administration guide[8]. To avoid problems with some features, please ensure nobody works with Signavio Workflow Accelerator while Signavio Process Manager is being updated (and vice versa).

This section contains step-by-step instructions on how to update Signavio Process Manager. To run the update, you need a .zip archive that contains the new version.

Please ensure nobody tries to use the system during the update.

1. Stop the Apache Tomcat server.

2. Create a backup of the database and the current program files. You will find the program files in the deployment folder (`webapps`) of the Apache Tomcat directory. You might also want to clean the Tomcat"work" and `conf/Catalina` folders.

3. **In case you are updating from version 8.6.0 or older**, please follow in addition to the instructions in the chapter *Install JDBC driver*.

4. Copy the update archive `SignavioEnterpriseX.X.X.zip` to the server.

5. Unpack the .zip archive. It contains the directory `files` and the new program files in it.

6. **In case you are updating from a version older than 9.7.0**, integrate the configuration files and the license file into the new installation files. You need to copy the files from several locations within the **tomcat's root directory** to the folder `conf` within the tomcat's root directory. Below, the tomcat's root directory is written as `$TOMCAT_DIR`. Accordingly, the target directory is `$TOMCAT_DIR/conf`.

   The following files need to be copied:

   - `$TOMCAT_DIR/webapps/ROOT/WEB-INF/classes/configuration.xml`

[8] https://docs.signavio.com/adminguide/workflow/en/signavio.html#compatibility-list

- `$TOMCAT_DIR/webapps/ROOT/WEB-INF/classes/configuration_signed.xml`

If available:

- `$TOMCAT_DIR/webapps/ROOT/WEB-INF/classes/simulation.xml`

If available and if you are using HTTPS, copy the following file and paste it into its original destination **after you moved the new program files to the new installation directory**:

- `$TOMCAT_DIR/webapps/solr/META-INF/context.xml`

In addition, go to `$TOMCAT_DIR/conf` to add the following information inside the `<Context>` tag of the file `context.xml`:

```
<Parameter name="configfile" value="${catalina.base}/conf/configuration.xml" />

<Parameter name="licensefile" value="${catalina.base}/conf/configuration_signed.xml" />

<Parameter name="configfile.simulation"
value="${catalina.base}/conf/simulation.xml" />
```

7. Delete the old program files from `$TOMCAT_DIR/webapps`.
8. Move the new program files from the directory `files` to `$TOMCAT_DIR/webapps`. Check if the system user that runs the Tomcat has access to newly copied files. If this is not the case, update the corresponding permissions accordingly.
9. Start the Apache Tomcat server.

Important: Never stop the Tomcat server while data migrations are running after an update, even if parts of the application can't be used due to configuration errors. To check if a data migration is running, check the logs of the Tomcat. If you are using a Tomcat cluster, only start one of the Tomcat servers and wait until all migrations have finished.

10. Use the *Diagnostics tool* (page 27) to ensure the system is configured correctly.
11. Delete the archive `SignavioEnterpriseX.X.X.zip` and the `SignavioEnterpriseX.X.X` folder.

Should the appliance of the update fail on any purpose, exchange the database, program and runtime files with the backups to be able to continue work with the saved data.

Hint: Since version 8.3.0 the search index will be recreated after each update (not on each restart, only on updates). This may take several minutes. Please note that the search may not work in that time and you won't be able to access the content of the dictionary.

Chapter 6

Replacing the license file

To replace the license file (for example after ordering additional users or after receiving a new license file within an On-Premise subscription), please follow the instructions in this article.

Important: Before replacing the license file please **backup the database**. The replacement of the license file triggers irreversible changes in the database.

1. Stop the Apache Tomcat Server.
2. Navigate to the folder `$TOMCAT_DIR/conf/`.
3. Backup the old license file `configuration_signed.xml`.
4. Move the new license file into the folder.
5. Restart the Apache Tomcat Server.

Chapter 7

Configuring Signavio Process Manager as a user

Signavio Process Manager provides extensive configuration options to adjust its functionality to the unique needs of your organization.

For detailed instructions on how to configure Signavio Process Manager as a *workspace administrator* user, read https://editor.signavio.com/userguide/en/workspace_admin/index.html.

Chapter 8

Support

If any problems occur while installing, maintaining or running Signavio Process Manager, save the log file of the Tomcat server and send it - together with further details - to the Signavio support.

Email: support@signavio.com

Phone: +49 30 856 21 54 - 21 (Monday to Friday, 9am to 5pm, CET)