

tts performance suite

MANUAL



Imprint

Copyright © tts Knowledge Products GmbH. All rights reserved.
Server Installation Manual – tts performance suite 2021 R2
3rd September 2021

Table of Contents

General Information.....	1
1.1 Introduction.....	1
1.1.1 Objective.....	1
1.1.2 Target audience	1
1.1.3 Prerequisites.....	1
1.1.4 Contact	1
1.1.5 Structure	1
1.1.6 Conventions.....	2
1.1.7 Your feedback is highly welcome.....	3
1.2 Application description	3
1.3 System overview and deployment scenarios.....	3
System requirements.....	5
2.1 Hardware and Software requirements	5
Preparations.....	6
3.1 License	6
3.2 Database	6
3.2.1 JDBC Driver	6
3.2.2 Microsoft SQL Server.....	7
3.2.3 Oracle Database.....	7
3.3 File system.....	7
3.4 Solr installation.....	8
3.5 MinIO installation	9
3.5.1 Installation on Windows	9
3.5.2 Installation as a Linux Service	9
Installation step-by-step.....	13
4.1 Checklist.....	13
4.2 Deployment of the Solr core.....	13
4.3 Installation environment.....	14
4.3.1 Set up installation environment.....	14
4.3.2 Configuring the tts performance suite using application-config.properties..	14
4.4 Data source.....	18
4.4.1 Creating a data source with Apache Tomcat.....	18
4.4.2 Escaping special characters	20
4.5 Application server settings.....	21
4.5.1 JVM settings	21
4.5.2 URL encoding.....	21
4.5.3 Cookie Processor	21

4.6	MinIO configuration.....	21
4.6.1	Install MinIO	21
4.6.2	Create a bucket	22
4.6.3	Set the URL to the repository	22
4.6.4	Set the credentials for MinIO.....	22
4.6.5	Set the MinIO endpoint	23
4.7	Deployment.....	23
4.8	Post installation steps.....	24
4.8.1	Initialization of the database schema	24
4.8.2	First login as administrator	26
4.8.3	Next steps.....	29
4.9	Troubleshooting.....	29
4.9.1	The server does not start up	30
4.9.2	Database is unavailable.....	30
4.9.3	Login fails	31
Migrating from previous versions.....		32
5.1	General remarks.....	32
5.2	Updating from 2021 to 2021 R2.....	32
5.2.1	Secure connection (SSL/TLS) required	32
5.2.2	Cookies.....	32
5.2.3	Solr 8.9.x required	33
5.2.4	New version of MinIO.....	33
5.2.5	MinIO configuration.....	33
5.2.6	Logging	34
5.2.7	XercesXMLSerializerFactory.....	34
5.3	Updating from version 2020 R2 to 2021	34
5.3.1	Solr configuration.....	34
5.3.2	Solr 8.7.x required	34
5.3.3	Redis Cache	35
5.3.4	New SAML implementation.....	36
5.3.5	Workflow synchronization has been removed.....	36
5.3.6	Tomcat 8.5 not supported anymore	37
5.4	Updating from version 2020 to 2020 R2	37
5.4.1	Creator.....	37
5.4.2	Logging Service.....	37
5.4.3	Other parameters	37
5.5	Updating from version 2019 R2 to 2020	37
5.5.1	Migration of data stores	37
5.5.2	MinIO	38
5.6	Updating from version 2019 to 2019R2	38
5.7	Updating from version 2018 R2 to 2019	38
5.7.1	Persistence	38
5.7.2	Quick Access Performance Optimizations	39
5.8	Updating from version 2018 to 2018 R2	40

5.9	Updating from version 2017 R2 to 2018	40
5.10	Updating from version 2015 R2 to 2016 R2.....	45
5.10.1	Epss and Performance Support Categories.....	45
5.11	Updating from version 2014 R2 to 2015 R2.....	45
5.11.1	Password Management for more Security.....	45
5.12	Updating from version 2013 R2 to 2014	48
5.12.1	Solr Search Engine	48
5.12.2	New mandatory application properties.....	48
5.12.3	New optional application properties	49
5.13	Updating from version 2013 to 2013 R2	52
5.13.1	Curator supports additional workflow functions.....	52
5.14	Updating from version 2012 R2 to 2013	53
5.14.1	WebAccess.....	53
5.14.2	WebAccess with more flexible Windows-based SSO configuration.....	54
5.14.3	Curator & WebAccess: Combine attributes in LDAP groupMembership	54
5.14.4	Curator supports definition of more mime-types for zip archives	54
5.15	Updating from version 2012 to 2012 R2	56
5.15.1	Configuration of the WebAccess.....	56
5.16	Updating from version 7.1 to 2012.....	57
Appendix.....		58
6.1	Properties service	58
6.2	Data service	66
6.2.1	JNDI data source.....	66
6.3	Store service	67
6.4	User service for authentication and authorization.....	69
6.4.1	Login modules.....	69
6.4.2	LDAP authentication	72
6.4.3	Single-Sign-On (SSO).....	79
6.4.4	Single-Sign-On (SSO).....	80
6.4.5	SAML Single-Sign-On.....	89
6.5	Logging	95
6.6	Version control and Workflow service.....	97
6.6.1	Version control	97
6.6.2	Workflow service.....	98
6.7	Cache service.....	100
6.8	Repository service	103
6.9	Notification service	104
6.10	Language service	107
6.11	Template service	108
6.12	Scheduler service.....	110

6.13	Configuration service.....	110
6.14	Feature service.....	112
6.15	Miscellaneous parameters	113
6.16	Search service	115
6.16.1	Solr Search Service.....	115
6.16.2	Search parameter	116
6.16.3	Highlighting parameter.....	117
6.17	Security configuration	119
6.18	Dashboard.....	122
6.18.1	License and user rights.....	122
6.18.2	Configuration of the Piwik connection.....	123
6.18.3	Specific error pages.....	123
6.19	User Import Process	123
6.20	Overview about the login modules	125
6.21	Adapt message for incompatible server & Producer	125
6.22	Configure WebSocket Support for Apache Webserver with AJP	126
6.22.1	Missing WebSocket Support In AJP	126
6.22.2	How to configure Apache Webserver?.....	126
6.23	Security Recommendations.....	127
6.23.1	HTTP Strict Transfer Security.....	127
6.23.2	HTTP Request Smuggling.....	129

General Information

1.1 Introduction

1.1.1 Objective

This document describes the requirements of the tts server components, named Curator and WebAccess, and their installation process.

The goal of this document is to provide an understanding of the server configuration and to guarantee its successful installation.

1.1.2 Target audience

System administrators, developers and all interested parties

1.1.3 Prerequisites

It is expected that system administrators, developers and who else may be interested have good knowledge of

- Windows and/or Unix-based operating systems
- Administration and handling of database management systems (Microsoft SQL Server/Oracle)
- Administration and handling of application server Apache Tomcat
- Deployment of web applications (WAR, external web application)
- SQL, HTML, XML



Please read this installation guide completely and carefully!

1.1.4 Contact

- tts Support
Phone: +49 (0) 2 21 / 17 09 30 -110
Fax: +49 (0) 2 21 / 17 09 30 – 170
support@tt-s.com
- Application consultant
Application consultants are very experienced with the tts performance suite and can guide you through the installation process, providing professional solutions to match your requirements. If no application consultant has yet been assigned, please contact your key account manager.

1.1.5 Structure

The first chapter sheds light on the functionalities and technologies used in the tts performance suite and finished by presenting the System overview and deployments scenarios.

The next section points to the System requirements concerning hardware and software. They should be checked thoroughly and carefully.

The Preparations chapter contains information on the prerequisites to get the tts server running.

The Installation chapter guides you step-by-step through the deployment of the tts server components, including all necessary actions like creating database connections or defining server configuration. Post installation steps deals with initial administration tasks. This chapter closes with a Troubleshooting section, which provides solutions for common problems.

Due to the fact that the tts server is a highly complex application, many configuration options are provided. A variety of other settings are explained in detail in the Appendix. Each property or parameter of the application's services is listed with its name, a description and the possible values.

1.1.6 Conventions

1.1.6.1 Symbols

To highlight important information on the one hand, and "nice to know" details on the other, the following icons are used:

 Attention

 Hint or note

 Tip

1.1.6.2 Abbreviations

TTG	TT Guide
TTKF	TT Knowledge Force
DB	Database
JNDI	Java Naming and Directory Interface
LDAP	Lightweight Directory Access Protocol
SOLR	Solr Search Engine
SSO	Single-Sign-On
WAR	Web Archive

1.1.6.3 Variables

 Variables are marked with a leading '\$'

\$TTPS_HOME	installation directory of tts performance suite
\$TOMCAT_HOME	installation directory of Apache Tomcat
\$SOLR_INDEX	index directory of Apache Solr for the tts performance suite

1.1.7 Your feedback is highly welcome

tts welcomes your feedback concerning the quality and usefulness of this manual. Your comments and suggestions will be considered as valuable input for future revisions of this manual.

- Found an error? Please let us know where.
- A topic is not described clearly enough? Please let us know which one.
- Need more information? On which topic?
- Something doesn't work for you? Please let us know so we can provide additional examples.

Please feel free to send us your feedback: support@tt-s.com. We appreciate your help!

1.2 Application description

The tts server consists of two components: the Curator and WebAccess. Representing a production and delivery platform, the Curator provides author functionalities, whereas the WebAccess makes the content accessible to the end users.

The server components have been developed in a platform-independent way, using technologies like Java, HTML, JavaScript, CSS, XML, and XSLT (to create platform-independent document types like RTF or PDF). There are only a few components that are platform-dependent, like the 'recorder' software or the 'QuickAccess for Desktop Applications' online-help-system, due to the technology used in them.

As a result of the server component's architecture, it can take little effort to integrate the tts server in existing environments.

1.3 System overview and deployment scenarios

This chapter offers a short outline of the tts Server system environment in two different deployment scenarios.

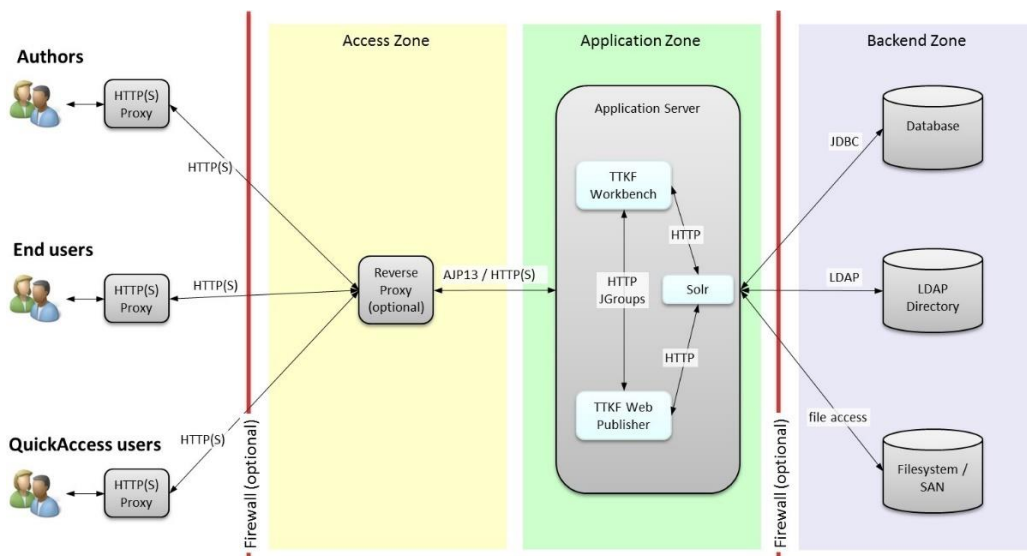


Figure 1

The first scenario (Figure 1), which is the minimum way in practice, involves one application server hosting both Curator and WebAccess, along with the Solr web application. The backend provides a database, file system (as repository), and an optional LDAP directory for authentication and authorization purposes. End users access the application via HTTP(S).

For performance and security reasons, we strongly recommend a deployment scenario using a reverse proxy.

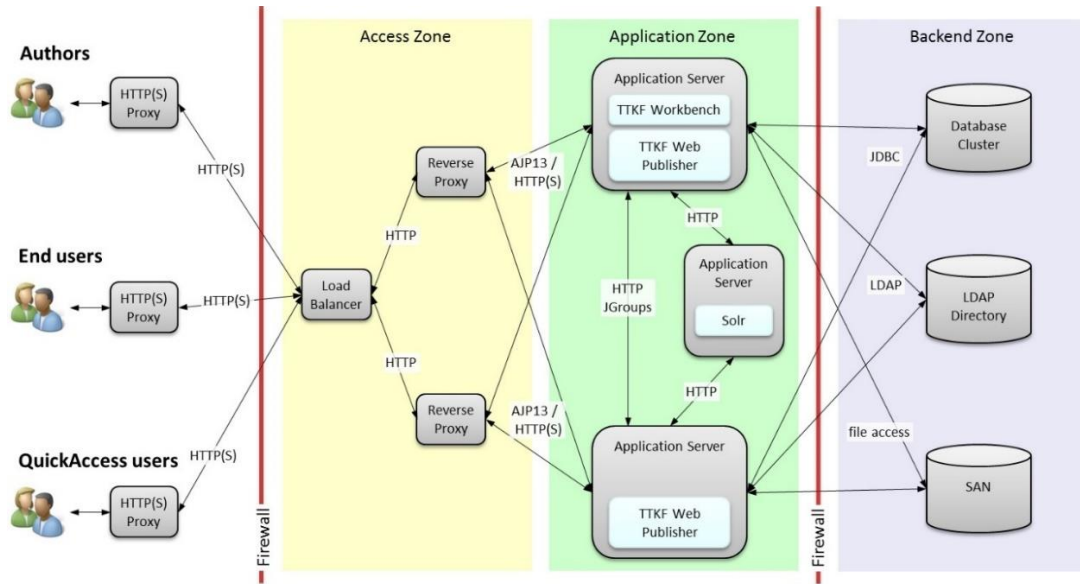



Figure 2

The second scenario (Figure 2) differs from the first one by being more scalable and dynamic, thanks to using three application servers plus a load balancer that distributes all requests among the reverse proxies. One application server hosts the Curator and one (Preview-) WebAccess. The other server(s) run(s) the main WebAccess. The third one provides with Solr web application the search engine.

Both scenarios, minimum and recommended, are field-tested concepts. The usage depends on the customer's requirements. Nevertheless, we recommend the deployment including a reverse proxy in any case. Moreover, if there is a huge number of end users expected, an installation in a clustered environment is advised, since this scenario offers more flexibility, better scalability, and higher performance.

 When using AJP, please read the chapter Configure websocket support for Apache Webserver with AJP in the appendix.

System requirements

2.1 Hardware and Software requirements




For the current hardware and software requirements of tts server, please refer to the system requirements.

Preparations

3.1 License

Before you start the installation process, a valid license for the tts server is required. Please request the license file via your professional service consultant or tts Support (support@tts.com). Please provide the MAC addresses as well as the IP addresses of the servers on which the components are going to be installed.

 Both components, Curator and WebAccess, need their own license file. If you plan to install the applications on separate machines, the MAC and IP address of each server has to be provided.

3.2 Database


To store data persistently, the tts server components require a relational database. Therefore, a new database must be created in your database management system (DBMS).

A database user is needed to access the database. The user must be granted particular permissions for the database, explained later, as well as permissions for Microsoft SQL Server and Oracle.

The user will be configured in the local application context XMLs. Integrated Authentication utilizing an AD user logged into Tomcat is also supported.

With regard to the server configuration, the following pieces of information are required:

- Host name or IP address of the database server
- TCP port of the database service
- Name of the database
- Names and passwords of the database users

 UTF-8 should be used as character set to store data in the database. If the database is used primarily for languages with a Unicode range higher than U+0800 (see www.unicode.org/charts for more information), a UTF-16 character set should be used.

3.2.1 JDBC Driver

Currently, the following JDBC drivers, as provided by the database vendors, are supported:

- Microsoft SQL Server JDBC Driver:
6.4.0+
7.2.0+
- Oracle JDBC Driver:
latest, compatible jdbc driver

We recommend using a driver that matches the database version, particularly for Oracle, as well as the installed Java version.

3.2.2 Microsoft SQL Server

The database user must have the permission to connect to the newly created database. This is usually done during user creation by setting the standard database. If not, access to the database has to be granted separately via the user properties.



The database user should be granted dbo rights, since he needs to create, alter and delete tables, procedures, functions, indices and so on.



Be aware that the port of the database might be set dynamically for a named instance. It may therefore deviate from the default value 1433.

3.2.3 Oracle Database

In Oracle, a tablespace must be created first. The related data file should have a size of at least 50 MB. To automatically assign more disk space, the "AUTOEXTEND" option must be activated. Otherwise, the database cannot store additional data once the space limit is reached.

The database user must be granted the following permissions:

```
SYSTEM PERMISSION: ROLE: CONNECT
SYSTEM PERMISSION: CREATE SESSION
SYSTEM PERMISSION: CREATE TABLE
SYSTEM PERMISSION: CREATE SEQUENCE
SYSTEM PERMISSION: CREATE PROCEDURE
```

3.2.3.1 Oracle 18c

In Oracle 12c an optimizer which can produce wrong query results in the tts performance suite was introduced. It should be disabled by these two parameters:

```
alter system set optimizer_adaptive_statistics=false
alter system set optimizer_adaptive_plans=false
```

The database user must also be granted unlimited quota for the created tablespace.



Upon request, tts Support can provide you with SQL scripts that can be used to generate the required tablespace and user, and to set the specific rights for the user.

3.3 File system

The following directories are required for the tts server components. One for the repository to save the document and guide contents. The other one is used as a data store for runtime files. The user of the application server needs complete access to those directories to read, write, execute and delete files.

Sample File Structure

```
$TTPS_HOME/repository/guide
$TTPS_HOME/store
```



Please refer to the hardware requirements section for information on the hard disk space that is required for both the repository and the data store.



Please make sure the created directories are accessible to all installed instances of the tts server. This also holds true if the instances are running on different machines.

3.4 Solr installation

The tts performance suite infrastructure uses Apache Solr™ for indexing and searching.

Apache Solr™ is a web application available from the Apache Software Foundation. Please install Solr according to their installation instructions:

https://solr.apache.org/guide/8_9/installing-solr.html



The tts performance suite requires a Solr version 8.9.x.



Please use a dedicated Solr instance for the tts performance suite infrastructure.



Please make sure that Solr is always started before Apache Tomcat.



For Windows servers we recommend to install the Solr as a service using NSSM <https://nssm.cc>. A detailed installation can be provided by Professional Services.



You will need the location of the Solr configuration files, which contain the index. We will refer to that location as \$SOLR_HOME. The exact location may differ, depending on your Solr installation, as explained in https://solr.apache.org/guide/8_9/solr-configuration-files.html.



If you prefer to use a custom location for your index, you will need to configure Solr accordingly, and you will need to copy the file solr.xml from the default location into your dedicated \$SOLR_HOME.

You will need the following properties for the tts Server Components installation:

- \$SOLR_HOME, e.g., C:\solr\server\solr
- Solr's URL and Port, e.g., http://127.0.0.1:8983/solr

After installing Solr, you can verify the installation as follows:

- Start Solr, if necessary
- Open the Solr main URL in your browser, e.g., http://127.0.0.1:8983/solr
- Check that the Solr dashboard shows the expected version.

3.5 MinIO installation

The tts performance suite infrastructure now uses MinIO as interface for the filesystem. MinIO provides a S3-like access to common filesystems, giving advantage of the S3 protocol on-premise, too.



For more Information on MinIO please refer to the official MinIO documentation <https://min.io/>

3.5.1 Installation on Windows

Download the currently supported MinIO version (please see the System Requirements) from the MinIO download archive: <https://dl.min.io/server/minio/release/windows-amd64/archive/> and rename the downloaded file to minio.exe.

The directory you have set for MinIO data in the configuration step will be referred to as <MINIO_DATA>.

To start MinIO run the following command.

```
minio.exe server /path/to/folderContaining<MINIO_DATA>/
```

The MinIO server is now started with the default credentials. To change the credentials run the following command.

```
set MINIO_ROOT_USER=newKey
set MINIO_ROOT_PASSWORD=newSecretKey
```



For windows servers we recommend to install MinIO as a service utilizing NSSM <https://nssm.cc>. A detailed installation description can be provided by Professional Services.



Please make sure that MinIO is always started before Apache Tomcat.

3.5.2 Installation as a Linux Service

The following steps will install MinIO as a SystemD backend service.

The following values are used throughout the setup and are initialized with default values which may be overridden.

```
MINIO_INSTALL_DIR=/usr/local/bin
MINIO_DATA_DIR=/var/minio
MINIO_ENV_FILE=/etc/default/minio
MINIO_GROUP_NAME=minio MINIO_USER_NAME=minio
MINIO_PORT=9000
BUCKET_NAME=bucket
```

Add a user and group to run minio.

```
groupadd $MINIO_GROUP_NAME
useradd -s /sbin/nologin -d /home/minio $MINIO_USER_NAME -g
$MINIO_GROUP_NAME
```

The next steps require these directories to exist.

```
mkdir -p $MINIO_INSTALL_DIR
mkdir -p $MINIO_DATA_DIR
mkdir $MINIO_DATA_DIR/$BUCKET_NAME
chown -R $MINIO_USER_NAME:$MINIO_GROUP_NAME $MINIO_DATA_DIR
```

Download the currently supported MinIO version (please see the System Requirements) from the MinIO download archive: <https://dl.min.io/server/minio/release/windows-amd64/archive/> and set the permissions:

```
wget https://dl.min.io/server/minio/release/linux-
amd64/archive/minio.RELEASE.<VERSION_NUMBER> -O
$MINIO_INSTALL_DIR/minio
```

```
chown -R $MINIO_USER_NAME:$MINIO_GROUP_NAME $MINIO_INSTALL_DIR
chmod u+x $MINIO_INSTALL_DIR/minio
chown -R $MINIO_USER_NAME:$MINIO_GROUP_NAME $MINIO_DATA_DIR
```

The basic configuration of the MinIO server, including the credentials, is contained in the *MINIO_ENV_FILE*. Replace the values for *MINIO_ROOT_USER* and *MINIO_ROOT_PASSWORD* with unique, random, secure values.

```
cat <<EOT > $MINIO_ENV_FILE
MINIO_VOLUMES="$MINIO_DATA_DIR".
MINIO_OPTS="--address :$MINIO_PORT"
MINIO_ROOT_USER=tempkeyid
MINIO_ROOT_PASSWORD=tempkey123456789
EOT
```

```
chown $MINIO_USER_NAME.$MINIO_GROUP_NAME $MINIO_ENV_FILE
```

With the above-mentioned environment variables, create this SystemD service file.


```

cat <<EOF > /etc/systemd/system/minio.service
[Unit]
Description=MinIO
Documentation=https://docs.min.io
Wants=network-online.target
After=network-online.target, cloud-final.service
AssertFileIsExecutable=${MINIO_INSTALL_DIR}/minio

[Service]
WorkingDirectory=${MINIO_DATA_DIR}
User=${MINIO_USER_NAME}
Group=${MINIO_GROUP_NAME}

EnvironmentFile=${MINIO_ENV_FILE}
ExecStartPre=/bin/bash -c "if [ -z \"\${MINIO_VOLUMES}\" ]; then
echo
\"Variable MINIO_VOLUMES not set in ${MINIO_ENV_FILE}\"; exit 1; fi"

ExecStart=${MINIO_INSTALL_DIR}/minio server \${MINIO_OPTS}
\${MINIO_VOLUMES}

# Let systemd restart this service always
Restart=always

# Specifies the maximum file descriptor number that can be opened by
this process
LimitNOFILE=65536

# Disable timeout logic and wait until process is stopped
TimeoutStopSec=infinity
SendSIGKILL=no

[Install]
WantedBy=multi-user.target
EOF

```

You can now enable and run the MinIO service using systemctl.

```

systemctl enable minio
systemctl start minio

```

You can use the standard commands from `systemctl` to manage MinIO.

```
systemctl start minio
systemctl stop minio
systemctl restart minio
systemctl status minio
```

The output can be retrieved from journal.

```
journalctl -u minio
```

Installation step-by-step

4.1 Checklist

Before you start deploying the tts performance suite server, please ensure that the following requirements are met:

- The installation files are available.
- The licenses for the components are available.
- Database and Database user(s) have been created.
- All required paths on file system are available.
- Details of distributed caching (Multicast-IP-address and port and so on).
- The application server is installed properly.
- The Solr search engine is installed, and \$SOLR_INDEX is configured.
- The MinIO Server is installed and configured



Before you start tts performance suite server for the first time, you are strongly advised to read the *Post installation steps* chapter.

4.2 Deployment of the Solr core



A Solr installation is required for this step. Please see *Solr installation*.

In this step, you deploy the preconfigured Solr core to your Solr installation.

1. Stop the Solr service, if it is running.
2. In the \$SOLR_HOME directory (see *Solr installation*), unzip the artifact *solr.zip*. This will create a new subdirectory *tts-server*.
3. Confirm the following directory structure:

```
$SOLR_HOME/core/core.properties
$SOLR_HOME/core/conf
$SOLR_HOME/core/conf/solrconfig.xml
$SOLR_HOME/core/conf/schema.xml
```

4. Optional: You can change the name of your Solr core by renaming the *tts-server*. In that case, please use the new name in the following instructions.
5. Start Solr.
6. Visit `http://[solrUrl]:[solrPort]/solr/#/core` to confirm Solr is running and the required core is present. E.g., `http://localhost:8983/solr/#/tts-server`



If you do not have the Solr configuration files, please consult your application consultant.

4.3 Installation environment

You need to configure the tts server components before you can deploy and start the application successfully. The basic configuration comprises **server properties**, **database connection**, and **file system paths**. Once these settings have been provided, the server can be run in default mode.

Additional settings may be necessary for services and features like user authentication and authorization, caching, or workflow and versioning. They are explained in detail in the appendix.

The actual configuration is carried out via a file named *application-config.properties*. Please note that the application has a fallback mechanism to load configuration parameters by scanning several external locations before using the system's default value:

1. Properties file with default name *application-config.properties*
2. Context parameters
3. JVM arguments

4.3.1 Set up installation environment

As best practice, we recommend setting up a simple installation environment to keep things clear. Therefore, just create the following folder structure:

```
$TTPS_HOME/apps
$TTPS_HOME/logs
$TTPS_HOME/repository
$TTPS_HOME/repository/guide
$TTPS_HOME/store
$TTPS_HOME/temp
```

The *apps* folder is there to store the single .WAR files of the Curator and the WebAccess.

The *logs* directory may contain generated log files, if you configure it that way. Especially if the application server does not unpack the .WAR files during deployment, we will need to configure the application to write logs in that folder.

The application's file repository will be setup within *repository*; runtime files are located in *store*. For temporary installation purposes, files may be stored in *temp*.

Of course, you may place these folders wherever you like, but please consider at least keeping the document content within *repository*, separate from runtime files in *store*.

At last, we place the server license file and the *application-config.properties* directly in *\$TTPS_HOME*.

4.3.2 Configuring the tts performance suite using *application-config.properties*

4.3.2.1 What is *application-config.properties*

The *application-config.properties* file is actually a simple text file with a *.properties* extension, serving - among other things - especially as a configuration mechanism for Java applications. Within this file, you can define application parameters using key-value-pairs separated by a '='.

Defining parameters is subject to simple rules:

- The file must have a *.properties* extension
- Parameter and value are separated with an '=' on a single line
- Parameter and value are valid
- Parameters for Curator match the following syntax:
ttkf.server.<servicename>.<propertyname> or
ttkf.integrator.<servicename>.<propertyname>
- Parameters for WebAccess match the following syntax:
ttkf.server.<servicename>.<propertyname> or
ttkf.accelerator.<servicename>.<propertyname>
- File paths must be separated with a slash "/" instead of backslash "\"

The following example shows a snippet of an *application-config.properties* file with a configuration for a SQL Server datasource:

```
# curator database configuration
ttkf.server.data.hibernate.default_schema=tts
ttkf.server.data.hibernate.dialect=
de.tts.bd.business.data.UnicodeSQLServerDialect

# webaccess database configuration
ttkf.accelerator.data.portal.hibernate.default_schema=tts
ttkf.accelerator.data.portal.hibernate.dialect=
de.tts.bd.business.data.UnicodeSQLServerDialect

# store directory
ttkf.server.store.base.directory=/ttkf/datastore
```

4.3.2.2 How to make *application-config.properties* accessible?

There exist two options to provide the application with the configuration file externally. Either you extend the class path so the application will find that properties file; or you directly define its path within the context file as a parameter.

For better maintenance, we suggest to configure the path to the location of *application-config.properties* directly in the corresponding context file.

To do so, add the *ttkf.server.properties* context parameter in the context file during deployment (see the deployment chapter):

```
...
<Parameter name="ttkf.server.properties" override="false" value="C:/tts performance suite/application-config.properties" />
...
```

4.3.2.3 Example of a minimum configuration

The following example illustrates the installation of the tts server components using an Oracle database.

Database connection

Initially, you configure the database connection. Since we want to use a data source, we just need to set hibernate dialect and default schema. The internal JNDI name to which the data source is bound is `java:comp/env/jdbc/TTKFDS`.

```
ttkf.server.data.hibernate.dialect=
de.tts.bd.business.data.UnicodeOracle12cDialect
ttkf.server.data.hibernate.default_schema=TTS
```

We define the connection properties for WebAccess as well:

```
ttkf.accelerator.data.portal.hibernate.dialect=
de.tts.bd.business.data.UnicodeOracle12cDialect
ttkf.accelerator.data.hibernate.portal.default_schema=TTS
```



See the *Database service* chapter in the appendix for more details.

Application properties

We begin by defining the context names of Curator and WebAccess. Since the context names can be chosen freely, it is necessary for intercommunication of the applications to know the context name of each other.

```
ttkf.server.properties.acceleratorContextPath = /webaccess
ttkf.server.properties.integratorContextPath = /curator
```

Next, we set the location of license file, the directory of the application server's logs and the super user IPs.

```
ttkf.server.properties.licenseFile=file:C:/tts/server.ttlk
ttkf.server.properties.applicationServerLogDirectory=C:/tts/logs
ttkf.server.properties.superuserIPs=127.0.0.1,192.168.85.48
```

Because only the Curator can re/index entities with Solr, the Curator and the WebAccess have to know each other's URL.

```
ttkf.server.properties.internal_acceleratorURL =
http://192.168.86.44:9980/webaccess
ttkf.server.properties.internal_integratorURL =
http://192.168.86.44:9980/curator
```



See the *Properties service* chapter in the appendix for more details.

Solr

This mandatory parameter defines how the Curator and WebAccess can reach the Solr server.

```
ttkf.server.search.server.url=http://127.0.0.1:8983/solr/tts-server
```



See the *Search Service* chapter in the appendix for more details.

Data store

Now we need to configure the base directory of the runtime files. So, we just define the location of our created folder \$TTPS_HOME/store.

```
ttkf.server.store.base.directory = C:/ttps/store
```



See the *Store service* chapter in the appendix for more details.

Guide

To store Guides, a Guide-Repository is needed. So, we just define the location of our folder \$TTPS_HOME/repository/guide.

```
ttkf.server.guide.repository.url=file:///C:/ttps/repository/guide
```



See the *Miscellaneous parameters* chapter in the appendix for more details.

Distributed cache

To get the distributed cache working properly we need to configure several parameters. Be sure to have information about the mode of the distributed cache (unicast/multicast) and the corresponding IP-addresses and ports ready.

We will configure multicast, which is the default mode.

```
ttkf.server.cache.peers.multicastGroupPort=45637
ttkf.server.cache.peers.multicastGroupAddress=228.8.18.9
```



See the *Cache service* chapter in the appendix for more details.

Repository

For the use of MinIO certain parameters are needed.

To authenticate with MinIO:

```
ttkf.server.repository.accessKeyId = {MinIO credentials}
ttkf.server.repository.secretKey = {MinIO credentials}
```

Configure endpoint of MinIO:

```
ttkf.server.repository.endpoint=https://www.my-minio-server.com
```

If an external endpoint is needed, configure as following:

```
ttkf.server.repository.endpoint.internal=http://localhost:9000
ttkf.server.repository.endpoint.external=https://proxy/
```

Configure repository location:

```
ttkf.server.repository.url=s3://bucket/repository
```



See the *Repository service* chapter in the appendix for more details.

4.3.2.4 How to escape special characters within the `.properties` file

The encoding of a `.properties` file is per definition *ISO-8859-1*, also known as Latin-1. All non-Latin-1 characters must be entered using *Unicode* escape characters, e. g. `\uHHHH`, where HHHH is a hexadecimal index of the character in the Unicode character set. A non-Latin-1 text file can be converted to a correct `.properties` file by using the *native2ascii* tool that ships with the JDK.

4.4 Data source

4.4.1 Creating a data source with Apache Tomcat

In Apache Tomcat, you can create a data source in a very simple way by adding it as a resource. As best practice, it is recommended to define this resource in both files `webaccess.xml` and `curator.xml`, located at `$TOMCAT_HOME/conf/Catalina/localhost`.

Since version 8.5, Tomcat provides an improved database connection pool (DBCP2). As an alternative, you can also use the Tomcat JDBC Connection Pool, for this refer to <https://tomcat.apache.org/tomcat-8.5-doc/jdbc-pool.html#Introduction>.

In the following step-by-step example, a data source pointing to an Oracle database as well as a SQL Server database will be generated.

1. Stop the Tomcat if it is running.
2. Open the `webaccess.xml` (similarly for `curator.xml`) in `$TOMCAT_HOME/conf/Catalina/localhost`, where resources can be defined.
3. Add your data source as a resource like below example.

Syntax:

```
<Resource
name="[JNDI-name]"
auth="[authentication type]"
type="[type of datasource]"
username="[name of db user]"
password="[password of db user]"
driverClassName="[class name of jdbc driver]"
url="[connection url]"
initialsize="[initial number of connections on startup of pool]"
minIdle="[minimum number of idle connections to be kept all time]"
maxTotal="[maximum number of connections at the same time]"
maxIdle="[maximum number of idle connections to be kept at all
time]"
timeBetweenEvictionRunsMillis="30000"
minEvictableIdleTimeMillis="60000"
testOnConnect="true"
testWhileIdle="true"
validationQuery="select 1 from dual"
validationInterval="30000"
suspectTimeout="60"
logAbandoned="true"
removeAbandonedTimeout="120"
removeAbandoned="true"
abandonWhenPercentageFull="60"
jdbcInterceptors="ResetAbandonedTimer;SlowQueryReport"
/>
```

Example (Oracle):

```
<Resource
name="jdbc/GTTKFDS"
auth="Container" type="javax.sql.DataSource"
username="TTS"
password="TTS" driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@localhost:1521:XE"
initialsize="10"
minIdle="10"
maxTotal="100"
maxIdle="50"
timeBetweenEvictionRunsMillis="30000"
testOnConnect="true"
testWhileIdle="true"
validationQuery="select 1 from dual"
validationInterval="30000"
suspectTimeout="60"
logAbandoned="true"
removeAbandonedTimeout="120"
removeAbandoned="true"
abandonWhenPercentageFull="60"
jdbcInterceptors="ResetAbandonedTimer;SlowQueryReport"
/>
```

4. Continue deploying the applications.

4.4.2 Escaping special characters

If it is necessary to use special characters within the tomcat configuration files, e.g., for passwords, the characters have to be xml escaped. A [free online conversion tool](#) can be used to convert the characters.

Incorrect example:

```
<Resource name=="jdbc/GTTKFDS"
username="TTS"
password="!"$%&/ ()=?β²"..">
```

Correctly escaped example:

```
<Resource name="jdbc/GTTKFDS" username="TTS"
password="!&quot;;&#167;$%&amp;/ ()=?&#223;&#178;"..">
```

4.5 Application server settings

4.5.1 JVM settings

The JVM of each server instance must meet at least the following memory settings and might be adapted to the environment's requirements:

- Maximum Java heap size: 1024 MB (-Xmx1024m)
- Initial Java heap size: 256 MB (-Xms256m)
- Maximum thread stack size: (-Xss256k)
- -XX:MaxMetaspaceSize=512m
Since Java 8 the metaspace actually does not need to be set. So, either ignore the flag or set it to an appropriate value. In case of memory limitations, the value needs to be increased.

Depending on the Java distribution UTF-8 might not be the default encoding. Thus, the following JVM parameter has to be set:

```
-Dfile.encoding=utf-8
```

4.5.2 URL encoding

To ensure that parameters in URLs are interpreted correctly (as UTF-8), the `useBodyEncodingForURI` attribute must be defined. This is done in the `server.xml` file of the application server, by adding `useBodyEncodingForURI` attribute with `true` as its value in the `http-conector`. The default value is `false`.

Example:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
redirectPort="8443" useBodyEncodingForURI="true" />
```

4.5.3 Cookie Processor

For Tomcat versions 8.5 and above, it is necessary to define the `LegacyCookieProcessor` inside the `context.xml` (or in `curator.xml` and `webaccess.xml`) in order to avoid problems with session cookies:

```
<CookieProcessor className=
"org.apache.tomcat.util.http.LegacyCookieProcessor" />
```

4.6 MinIO configuration

4.6.1 Install MinIO



A MinIO installation is required for this step. Please see *MinIO installation*.

For this migration guide, you need the following:


- The local address of the MinIO server, e.g., `http://localhost:9000`, `<MINIO_INTERNAL>`
- If a load balancer or proxy is used, the external address as configured in the load


balancer or proxy configuration, <MINIO_EXTERNAL>

- The MinIO data directory, <MINIO_DATA>
- The MinIO credentials, <MINIO_ROOT_USER> and <MINIO_ROOT_PASSWORD>
Set during MinIO installation.

4.6.2 Create a bucket


Create a folder <bucket>, e.g., *data* or *bucket*, within <MINIO_DATA>

 This is where the document repository and any stores you wish to keep behind MinIO will be.

 Do not use uppercase or invalid characters like "_" for the bucket name.

4.6.3 Set the URL to the repository

The root path (<path>) will be the location for documents, e.g.,
<MINIO_DATA>/bucket/repository/d5/86443033-22da-4489-9678-85d1348940e0/file.txt

 Note that <path> may not be empty.

Set the repository URL parameter in *application-config.properties*:

```
ttkf.server.repository.url=s3://<bucket>/<path>
```

Use the S3 URL format, for example:

```
ttkf.server.repository.url=s3://bucket/repository
```

4.6.4 Set the credentials for MinIO

The tts performance suite supports several methods for supplying credentials, the simplest of which is to provide the values in the *application-config.properties*. All methods are described here and the best method can be selected according to your requirements.

For each of the following methods you will need an access key (<MINIO_ROOT_USER>, sometimes called the Access Key ID) and a secret key (<MINIO_ROOT_PASSWORD>).

4.6.4.1 application-config.properties

Set the following parameters in the *application-config.properties*:

```
ttkf.server.repository.accessKeyId=<MINIO_ROOT_USER>
ttkf.server.repository.secretKey=<MINIO_ROOT_PASSWORD>
```

4.6.4.2 Environment variables

The credentials can be passed to the Server by setting the environment variables `AWS_ACCESS_KEY_ID` and `AWS_SECRET_KEY`:

```
export AWS_ACCESS_KEY_ID=<MINIO_ROOT_USER>
export AWS_SECRET_KEY=<MINIO_ROOT_PASSWORD>
```

4.6.4.3 System properties

These properties may also be set in Tomcat's `catalina.properties` file:

```
aws.accessKeyId=<MINIO_ROOT_USER>
aws.secretKey=<MINIO_ROOT_PASSWORD>
```

4.6.4.4 AWS profile

The connection to MinIO can also be configured via an AWS profile. This can be done automatically (via the [AWS Command Line Interface](#) and the `aws configure` command) or manually, as described here.

Create a file `.aws/credentials` in your home directory with the following content:

```
[default]
aws_access_key_id=<MINIO_ROOT_USER>
aws_secret_access_key=<MINIO_ROOT_PASSWORD>
```



On Linux: `~/.aws/credentials`

On Windows: `%USERPROFILE%\.aws/credentials`

4.6.5 Set the MinIO endpoint

Set the following parameters in `application-config.properties`:

```
ttkf.server.repository.endpoint.internal=<MINIO_INTERNAL>
ttkf.server.repository.endpoint.external=<MINIO_EXTERNAL>
```



Trailing slashes are not allowed.



External endpoint is necessary only when using a load balancer or proxy or when the internal endpoint is not accessible from the author machines running the Producer.



Make sure MinIO is directly accessible from the author machines, otherwise the upload of documents will fail.



HTTP requests (especially PUT) to MinIO from author machines have to be possible.

4.7 Deployment

For Apache Tomcat, the deployment of the Curator is described exemplarily step-by-step. All steps must be carried out for the WebAccess instances as well.

The example shows the deployment via a `.WAR` file. Usually, tts provides pre-configured `.WAR` files for both components, Curator and WebAccess.

1. Stop Tomcat if it is running.
2. Copy `curator.war` (from `$TTPS_HOME/apps`) to `$TOMCAT_HOME/webapps`.
3. Start Tomcat. Now the server will unpack the `.WAR` file into `$TOMCAT_HOME/webapps/curator`

4. Stop Tomcat.
5. Create the *curator.xml* file in the *\$TOMCAT_HOME/conf/Catalina/localhost* directory.
6. Configure the data source and the location of the *application-config.properties* within this context file.

Example:

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

<Context path="/curator" reloadable="false">

<ResourceLink type="javax.sql.DataSource" name="jdbc/TTKFDS"
global="jdbc/GTTKFDS" />

<Parameter name="ttkf.server.properties" override="false"
value="C:/tts/application-config.properties" />

</Context>
```

7. Restart Tomcat. The Curator should be accessible via the context */curator* now.
8. Proceed with the WebAccess.

4.8 Post installation steps

4.8.1 Initialization of the database schema

If the steps described in the previous chapters have been carried out, the application should now be properly configured and deployed, thus being ready to start by starting Apache Tomcat.



To initialize the database schema, a browser has to be started on the application server machine. If this is not possible and you thus have to use the browser on another computer, the IP address of that machine must be declared as superuser IP (property *superuserIPs* of Properties service).



The webpage for initializing the database is located at the following URL:
[http://\[server\]:\[port\]/\[contextCurator\]/site/install/install.do](http://[server]:[port]/[contextCurator]/site/install/install.do)

Start by entering the URL given above in the browser. A straightforward administration page appears, offering to test the database connection, (re-)create or update the database schema, or to import/export the entire database content.

4.8.1.1 Test database connection

To ensure that the database connection is working properly and that the database user has been granted all necessary permissions, the Curator setup provides a database connection test:

TTKF Workbench setup

Here you can find the main functions needed to setup the installation of TTKF Workbench.

Database Feature migration

Database connection

Test the connection

Run this test to make sure a connection to the database can be established and that all necessary rights have been granted.

Database schema

Initialize database schema (deletes all current data)

In its default configuration, this function will delete an existing database, create a new, empty database according to the current schema. If the server has been configured to create SQL scripts, those scripts are created in the appropriate folder. The SQL scripts must be run manually, and the database population procedure must also be triggered manually afterwards.

Initially populate database

Attention: The database must be empty when calling up this function.

This function populates an existing/empty database with initial data.

Shortly after clicking the Test the connection link, the result of the test will be displayed, representing successful tests in green, failed operations in red print.

TTKF Workbench setup

Here you can find the main functions needed to setup the installation of TTKF Workbench.

Database Feature migration

Database connection

Test the connection

Run this test to make sure a connection to the database can be established and that all necessary rights have been granted.

1. Checking settings:

Driver › Microsoft SQL Server JDBC Driver 3.0

Dialect › de.tts.bd.business.data.UnicodeSQLServerDialect

Connection URL › jdbc:sqlserver://localhost:1433;xopenStates=false;sendTimeAsDatetime=true;trustServerCertificate=false;responseBuffering=adaptive;packetSize=8000;loginTimeout=15;lockTimeout=-1;lastUpdateCount=true;encrypt=false;disableSTJDBC Driver;

Username ›

2. Connection to the database has been established.

3. Test SQL scripts are being executed:

Test function has been created.

Test procedure was executed successfully.

Test function was executed successfully.

Test table has been created.

Test procedure has been created.

All test objects have been deleted.

Database connection test has been completed successfully.

4.8.1.2 Initialize database

If you click the *Initialize database* link, the Curator creates the database schema as well as initial data. After a few moments, the login screen will then appear. A full restart of the tts server will be necessary to ensure that all services are running properly.

TTKF Workbench setup

Here you can find the main functions needed to setup the installation of TTKF Workbench.

Database Feature migration

Database connection

Test the connection

Run this test to make sure a connection to the database can be established and that all necessary rights have been granted.

Database schema

Initialize database schema (deletes all current data)

In its default configuration, this function will delete an existing database, create a new, empty database according to the current configuration. If the server has been configured to create SQL scripts, those scripts are created in the appropriate folder. The SQL scripts must be run manually, and the database population procedure must also be triggered manually afterwards.

Initially populate database

Attention: The database must be empty when calling up this function.

This function populates an existing/empty database with initial data.

When (re-)creating the database schema or importing database content, the previous data will be deleted completely. Please use this function very carefully!

The WebAccess does not require to initialize the database schema. It is therefore recommended to install and start up the Curator standalone before installing (and starting) the WebAccess.



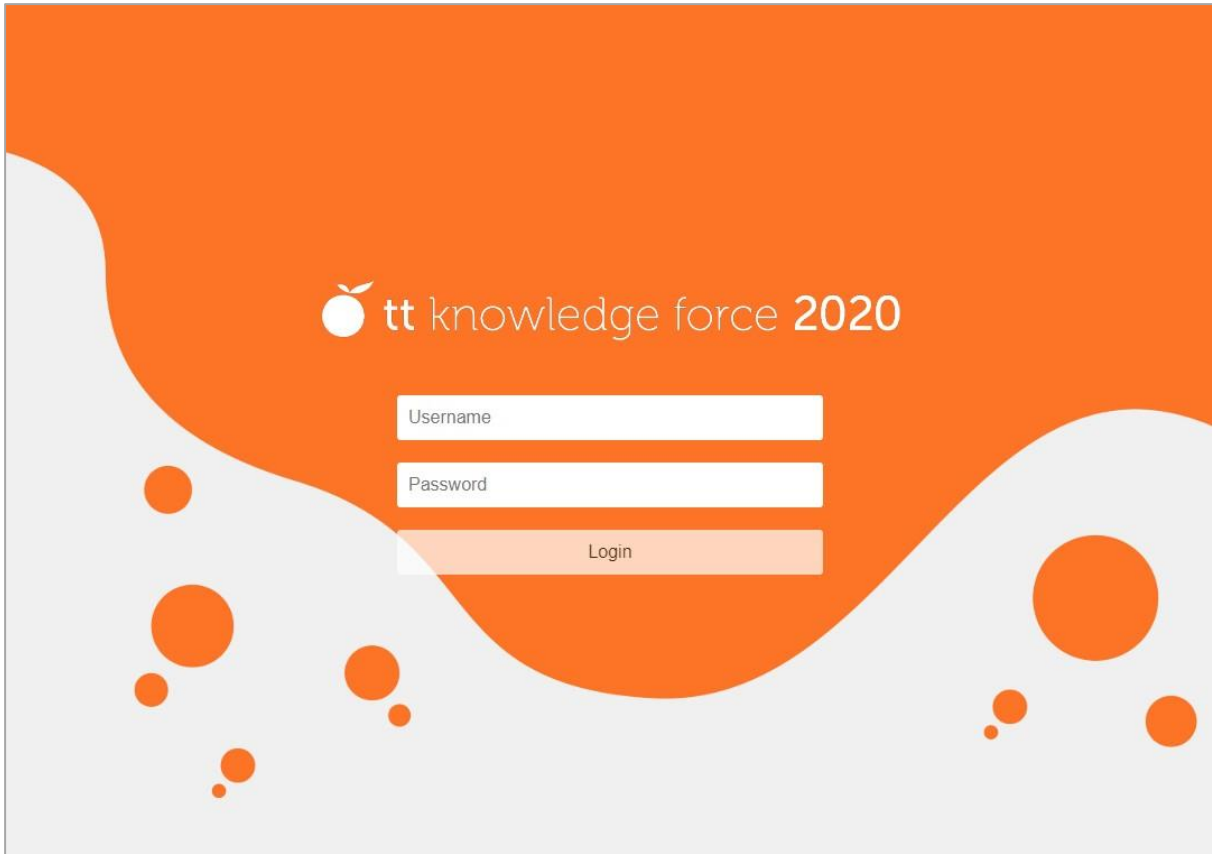
After initializing the database schema, the application should be restarted. Now the WebAccess may be started as well. For an easier analysis of possible problems, the log file of the application server should be saved before restarting the server or applications.

4.8.2 First login as administrator

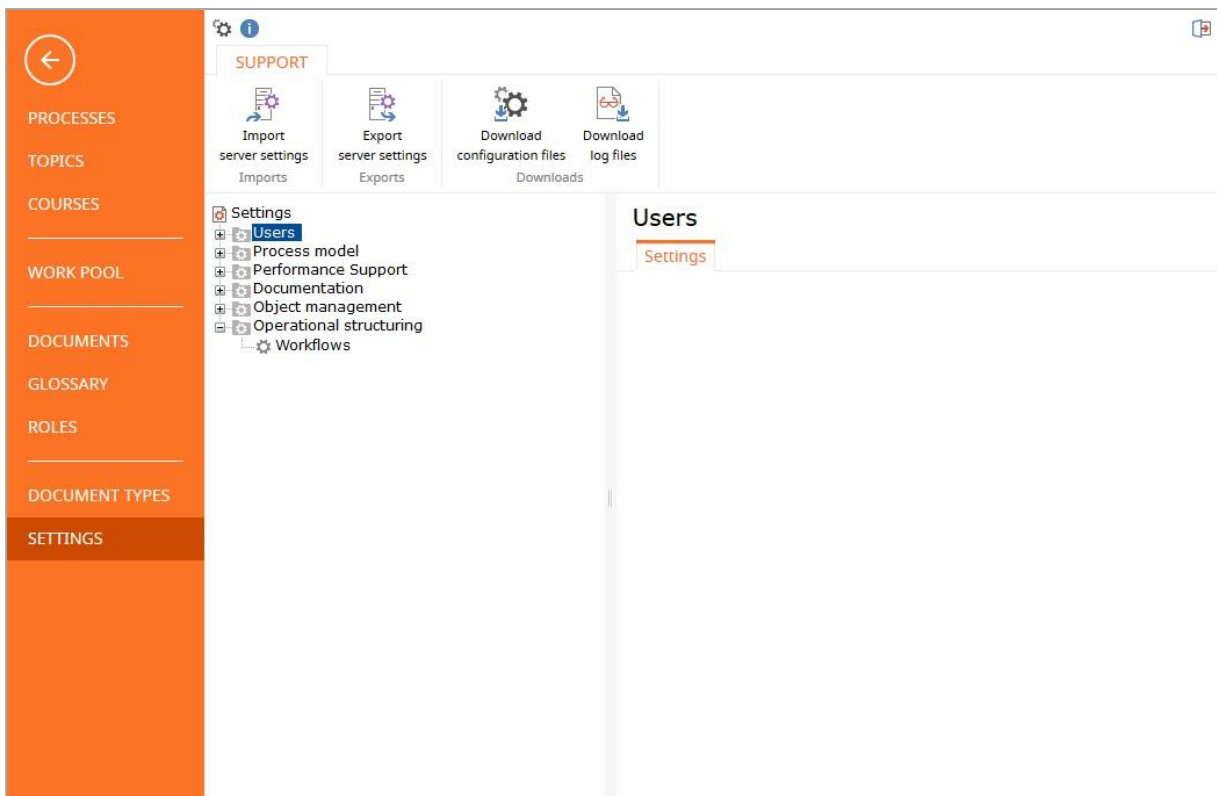
During database schema creation, a few initial data have already been created, like the administrator user:

Username	admin
Password	admin

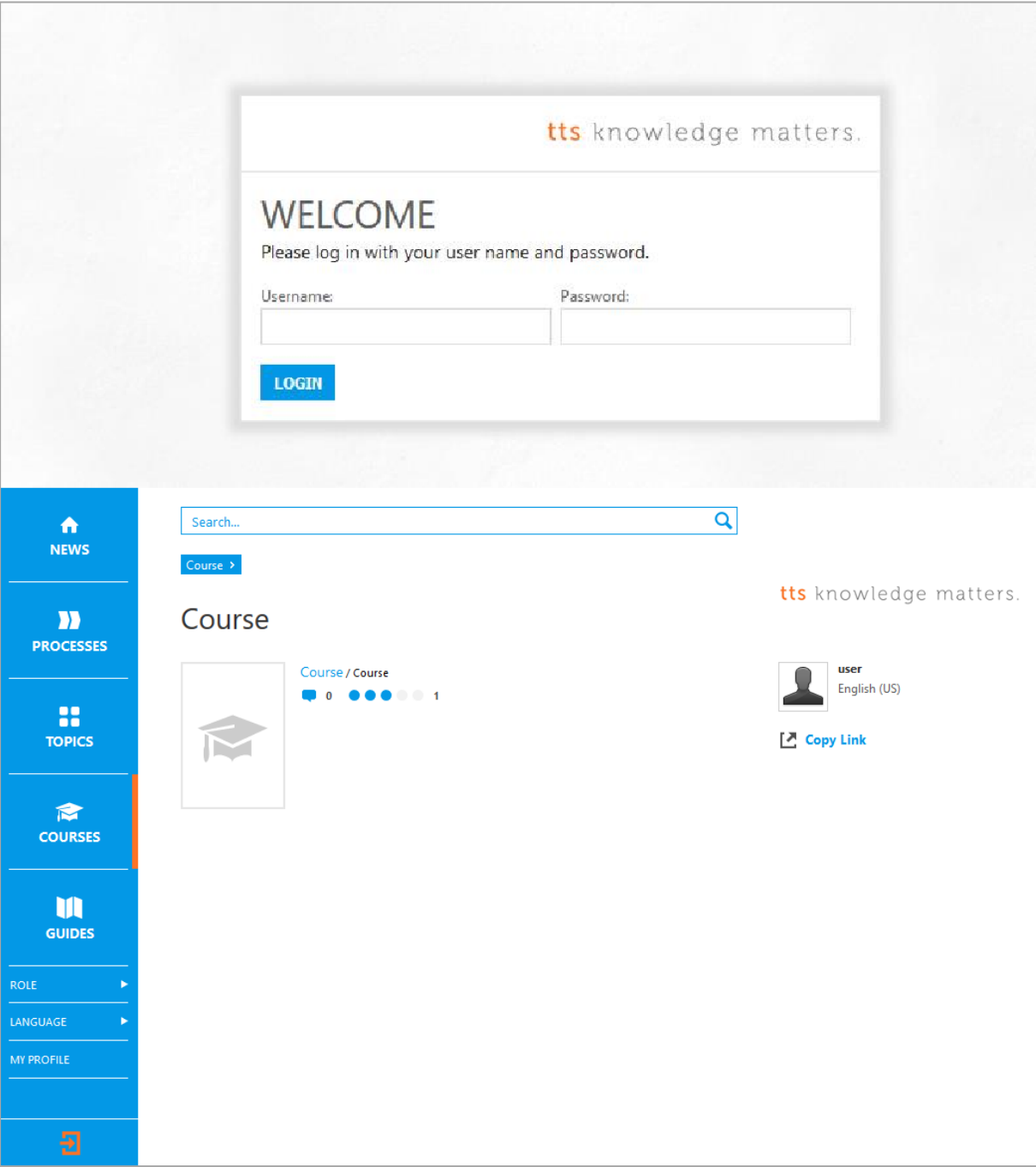
The first login is done by calling `http://[server]:[port]/[contextCurator]/` in the browser and entering the administrator's username and password.



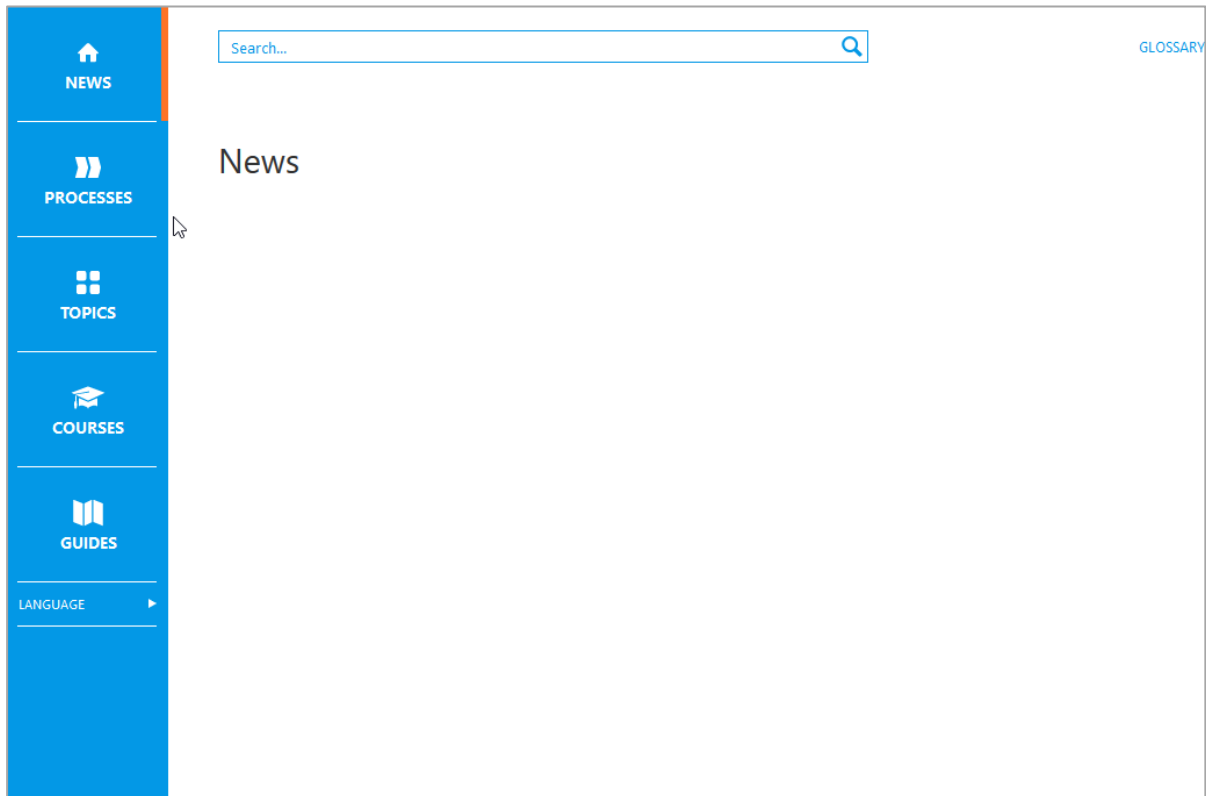
After logging in as administrator, the start page of the Curator will look like the following screen by default (depending on the license):



WebAccess with login required:



WebAccess without login required:



For security reasons, change the administrator password upon first login.

4.8.3 Next steps

The next steps for the administrator will usually encompass:

- Defining repositories and document types
- Assigning repositories to document types
- Adding users and maintenance roles
- Assigning authorizations to external users
- Defining a process model
- Managing custom properties
- ...




Please see the Administrator manual for more details on the steps outlined above.


4.9 Troubleshooting

The configuration and deployment of a complex application such as tts server sometimes results in errors. In this chapter, the most common issues and their solutions will be discussed.

Before you contact tts support, you are encouraged to have a look at this chapter, as an answer to the problem you are facing may be given here. In case the problem cannot be solved, tts support will be glad to help you.

Please note, however, that support activities are not covered by the maintenance fee if the problem results from a mistake the customer made during installation.

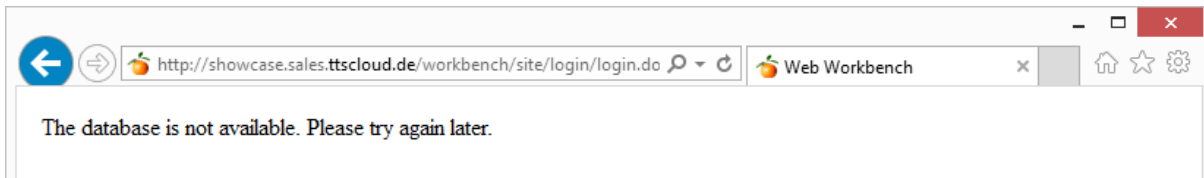
 In order for tts support to be able to help you in the best possible way, please attach log files and configuration files of the tts server to your support request.

 If the Curator is up and running, the log and configuration files can be downloaded from the Administration section (> Settings > Support > Server management). Otherwise, the log files of the application server should be attached instead.

4.9.1 The server does not start up

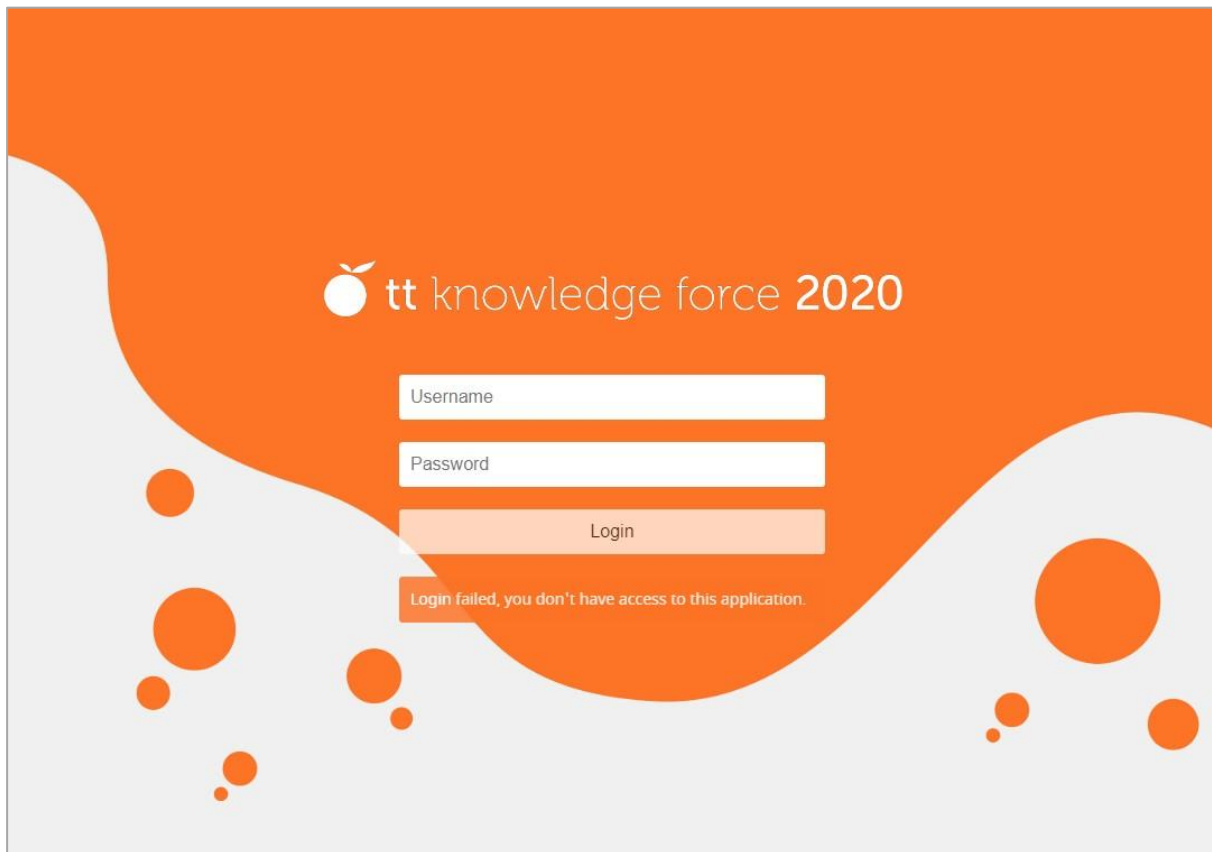
- Is the application server running correctly?
- Is the context path, if configured, pointing to the right directory (Apache Tomcat)?
- Is the license of tts server available and valid?
- Is the *application-config.properties* file available and configured properly?
- Is the data store defined in *application-config.properties* available?

4.9.2 Database is unavailable



- Is the database up and running?
- Can the database be accessed from the machine on which the application server is running?
- Has the port been set correctly?
- Is the specified data source available?
- Is the appropriate JDBC driver properly defined/stored on the application server?
- Is the data store defined in *application-config.properties* available?

4.9.3 Login fails



- Has the correct username and password been entered?
- Is the user already logged in?
- Has the maximum permissible number of logged-in users been reached?
- Has the license expired?
- Is this tts server addressing the right database (if pointing to a database initialized from another tts server, user login might fail)

Migrating from previous versions

5.1 General remarks

This chapter provides a short summary of the most important changes that were introduced in the major releases, focusing on installation and update routines as well as migration scenarios.

5.2 Updating from 2021 to 2021 R2

5.2.1 Secure connection (SSL/TLS) required

All external connections (e.g., from a browser to the Curator or from the QuickAccess to the WebAccess) must now be encrypted using TLS (previously called SSL). I.e., the following external URL:

```
ttkf.server.properties.external_integratorURL =
http://authoring.acme.com/curator
```

must be changed to:

```
ttkf.server.properties.external_integratorURL =
https://authoring.acme.com/curator
```

and the appropriate infrastructure configured, for instance using a Tomcat `<Connector>` with `SSLEnabled="true"` or using a Reverse Proxy.

The exception is for internal connections (e.g., between the Curator and the WebAccess). These do not require TLS and can continue using e.g., <http://localhost:8080/webaccess>.



Please refer to the Security Recommendations in the Appendix to secure your server.

5.2.2 Cookies

The following attributes are now required to a varying degree on all Cookies:

- `HttpOnly`
Always required, must not be configured
- `Secure`
Required for external requests
- `SameSite=None` or `SameSite=Lax`
Must be configured

5.2.2.1 Secure



This should be set automatically if the TLS infrastructure is set up correctly.

For Tomcat-based TLS, ensure the following attribute is set on the connector:

```
<Connector ... SSLEnabled="true" secure="true">
```

For Reverse Proxy-based solutions, the `X-Forwarded-Proto: https` header should suffice. If not, add the `secure="true"` attribute to the Connector as above.

5.2.2.2 SameSite

The attribute `sameSiteCookies` must be set in the `<CookieProcessor>` in the `server.xml` (or the equivalent `context.xml` file).

For the Curator:

```
<Context path="/curator" ...>
  ...
  <Parameter name="ttkf.server.properties" override="false"
value="..." />
  <CookieProcessor
className="org.apache.tomcat.util.http.LegacyCookieProcessor"
sameSiteCookies="None" />
</Context>
```



If not using the Creator, this may be set to `sameSiteCookies="Lax"`.

For the WebAccess:

```
<Context path="/webaccess" ...>
  ...
  <Parameter name="ttkf.server.properties" override="false"
value="..." />
  <CookieProcessor
className="org.apache.tomcat.util.http.LegacyCookieProcessor"
sameSiteCookies="Lax" />
</Context>
```

5.2.3 Solr 8.9.x required

The tts performance suite Server 2021 R2 requires Solr 8.9. Please:

- Install the latest version of Solr 8.9
- Copy the latest Solr configuration files (solr.zip)
Described in more detail in Solr Installation.

5.2.4 New version of MinIO

A new version of MinIO is recommended: [2021-09-03T03-56-13Z](#). Please check the system requirements for the latest recommended version.

5.2.5 MinIO configuration

The environment variables used for setting the MinIO credentials (`MINIO_ACCESS_KEY` and `MINIO_SECRET_KEY`) have new aliases and are deprecated. They should be replaced with the following:

- `MINIO_ACCESS_KEY` → `MINIO_ROOT_USER`
- `MINIO_SECRET_KEY` → `MINIO_ROOT_PASSWORD`

The variables used for password rotation (`MINIO_ACCESS_KEY_OLD` & `MINIO_SECRET_KEY_OLD`) are also deprecated. The new environment variables should simply be changed directly.



Please see here for more information on the new and old variables: https://docs.min.io/minio/baremetal/reference/minio-server/minio-server.html#envvar.MINIO_ACCESS_KEY.

5.2.6 Logging

Log4j1 was updated to log4j2 for the tts performance suite Server 2021 R2. The logging configuration can now be changed at runtime. However, the configuration has also changed significantly:

- log4j is now configured using a separate file. All parameters in `application-config.properties` starting with `ttkf.server.logging` have no effect anymore.
- The configuration syntax must be updated to conform with the log4j2 syntax. See the Logging chapter in the appendix or the official log4j2 documentation, that can be found here: <https://logging.apache.org/log4j/2.x/manual/configuration.html#Properties>

5.2.7 XercesXMLSerializerFactory

The following configuration parameters should be removed as they have no effect anymore.

```
ttkf.server.properties.XercesXMLSerializerFactory=...
ttkf.server.properties.xmlReaderClass=...
```

5.3 Updating from version 2020 R2 to 2021

5.3.1 Solr configuration

The Solr core (delivered in `solr.zip`) is now named `tts-server` by default. If you have not renamed it, you will likely need to update the URL in your `application-config.properties`:

```
ttkf.server.search.server.url=http://127.0.0.1:8983/solr/tts-server
```

where the host and port match your old configuration.



This may need to be done separately for both the Curator and the Web Access.

5.3.2 Solr 8.7.x required

The tts performance suite Server 2021 R1 requires a newer version of Solr 8.7. Please:

- Install the latest version of Solr 8.7

- Copy the latest Solr configuration files (solr.zip)
Described in more detail in Solr Installation.
- Update *application-config.properties* with the new default core name, tts-server, if required
Described above in the previous section.

5.3.3 Redis Cache

As of version 2021, a cache implementation for Redis is available. This section assumes a Redis installation is already set up.



Redis should only be installed on Linux servers. It is not recommended to install it on Windows servers. See <https://redislabs.com/ebook/appendix-a/a-3-installing-on-windows/a-3-1-drawbacks-of-redis-on-windows/> for more information.

The following two parameters are required for any Redis cache setup:

```
ttkf.server.cache.cacheProvider =
com.tts.serverfoundation.cache.redis.RedisCacheProvider
ttkf.server.cache.redis.endpoint=192.168.59.101:7000
```

Cache Provider

There are two separate cache provider implementations, one for a single redis instance (RedisCacheProvider) and one for a redis cluster (RedisClusterCacheProvider).

Endpoint

In addition, the redis endpoint must be configured. For a redis cluster, it is sufficient to provide the endpoint configuration for a single node in the cluster. The implementation will handle the detection of the remaining nodes.

The following parameters are optional:

```
ttkf.server.cache.redis.ssl=true
ttkf.server.cache.redis.user=username
ttkf.server.cache.redis.password=password
ttkf.server.cache.redis.connectionTimeout=10000
ttkf.server.cache.redis.maxAttempts=5
```

SSL

If set to true the communication with Redis will be TLS-secured (defaults to false).



The Redis installation must be configured accordingly to support encrypted communication

Username and Password

The Redis instance/cluster can be accessed using username and password if configured accordingly. It is possible to use username and password or just a password. By default, neither username nor password are used. See the Redis documentation for details.

Connection timeout and max attempts

Connection parameters for accessing a Redis instance/cluster from the tts performance suite. The defaults are 2000 ms for the timeout and 5 maximum connection attempts.



It is strongly recommended to configure the Redis instance to use TLS and to secure it by username and password

5.3.4 New SAML implementation

As of this version, a new SAML implementation is available for the Webaccess. The new implementation supports multiple IDPs and its configuration can be changed at runtime.



We will be happy to help you set up the new implementation. Please contact your professional services consultant if interested.

5.3.5 Workflow synchronization has been removed

The mechanism to synchronize the custom property *workflowstatus* (propertydefinition) with corresponding workflow is removed in tts performance suite 2021. This synchronization has been introduced for migration and convenience purposes and is not necessary anymore. Now the workflowstatus is controlled by workflow engine exclusively. In case the former custom property *workflowstatus* still exists, it will behave like a normal custom property and should be deleted manually during update procedure.

Update steps after migration to tts performance suite 2021

- If existing, delete property definition *workflowstatus* in Settings → Object definitions → Document
- Remove following configuration parameters in application-config.properties:

```
# no effect
ttkf.server.workflowService.metaAttribute=workflowstatus
ttkf.server.workflowService.importAttribute=workflowstatus
```

- In case the document excel import is used, the configuration of *workflowstatus* has changed. The parameter value *workflowstatus* is now type *property* instead of *systemproperty*.

```
<extractor name="workflowstatus">
  <set-parameter name="column" value="Q"/>
  <set-parameter name="property" value="workflowstatus"/>
</extractor>
```

- Check existing customizings for involvement of custom property *workflowstatus*, e.g. a custom property filter based off *workflowstatus* for documents.

5.3.6 Tomcat 8.5 not supported anymore

As of version 2021, only Tomcat 9 is supported. Customers running the server with Tomcat 8.5 must migrate to Tomcat 9!

5.4 Updating from version 2020 to 2020 R2

5.4.1 Creator

The parameter `ttkf.server.properties.external_integratorURL` is required for creating user generated content with the Creator. It has to be set in the `application-config.properties` of the WebAccess:

```
ttkf.server.properties.external_integratorURL =
https://myserver/curator
```

5.4.2 Logging Service



The default logging configuration has changed. See *Logging Service* for details on new default logfile locations and configuration possibilities.

5.4.3 Other parameters

There are 4 new parameters that can be configured in the `application-config.properties`:

- `ttkf.server.otp.expiration.seconds` (see Miscellaneous parameters)
- `ttkf.server.content.security.policy.restrictSVG` (see Miscellaneous parameters)
- `ttkf.server.content.security.policy.restrictXML` (see Miscellaneous parameters)
- `ttkf.server.properties.reject.user.agents.patterns` (see Properties Service)

5.5 Updating from version 2019 R2 to 2020

5.5.1 Migration of data stores

The configuration parameters for data store have changed in version 2020. The values for keys ending with `.url` are now given as a file- or S3-URL, depending on the location of the stores. The *(optional)* steps must only be completed if they were already configured.



If the store is in a folder managed by MinIO, it should be accessed using the S3 protocol.

This step-by-step guide uses filesystem examples. If the store is in a directory managed by MinIO (i.e., next to the repository), use an S3-URL. In other words, `C:/datastore/spool` would become `s3://path/to/spool`.



If using filesystem-based URLs, the folders must already exist.



If using S3-URLs, existing data has to be moved manually during migration.

- Rename the base directory of the store (**required**)
`ttkf.server.store.store.base.directory >`
`ttkf.server.store.base.directory`
- Remove the index store parameter if present:
`ttkf.server.store.store.index.directory`
- Rename the following store parameters and change the paths to URLs, if configured (optional)

<code>ttkf.server.store.store.config.directory</code>	<code>ttkf.server.store.config.url</code>
<code>ttkf.server.store.store.cre.directory</code>	<code>ttkf.server.store.cre.url</code>
<code>ttkf.server.store.store.image.directory</code>	<code>ttkf.server.store.image.url</code>
<code>ttkf.server.store.store.spool.directory</code>	<code>ttkf.server.store.spool.url</code>
<code>ttkf.server.store.store.report.directory</code>	<code>ttkf.server.store.report.url</code> <i>The report store cannot be an S3 path.</i>

- Configure the guide repository (**required**)
 - Rename
`ttkf.server.guidibase.rootdir >`
`ttkf.server.guide.repository.url`
 - Change directory path to a valid URL e.g.,
`ttkf.server.guide.repository.url=`
`file:///C:/datastore/guidibase`

5.5.2 MinIO



See *MinIO installation* for details on how to install MinIO.



See *MinIO configuration* for details on how to configure MinIO.

5.6 Updating from version 2019 to 2019R2

- Java needs to be migrated to JDK 11
- JDK 11 needs to be set in the Windows System Variables for Solr to function correctly
- Java needs to be configured in Tomcat

5.7 Updating from version 2018 R2 to 2019

5.7.1 Persistence

In release 2019 the persistence layer was heavily updated and optimized. With these changes some configuration adjustments are necessary to take.

Hibernate Dialects

The following hibernate dialects are now supported for tts performance suite 2019:

```
de.tts.bd.business.data.UnicodeOracle12cDialect
de.tts.bd.business.data.UnicodeSQLServerDialect
de.tts.bd.business.data.Oracle12cDialect
de.tts.bd.business.data.SQLServerDialect
```

Please be aware of which dialect to choose, either unicode or non-unicode.

JDBC drivers

The update of the current jdbc driver to these versions is required!

- Oracle
latest, compatible jdbc driver
- MS SQL Server
6.4.0 (mssql-jdbc-6.4.0.jre8.jar)
7.2.0 (mssql-jdbc-7.2.0.jre8.jar)

5.7.2 Quick Access Performance Optimizations

In 2019 several changes were made to optimize the performance of Quick Access.

Highlighting

The ShowExcerpts tag cannot be used for quick access customization any more, because the quick access search does not provide excerpts, due to performance reasons.

Count of groups and rows of search query

The count of requested groups (250) and rows (10.000) was decreased by default to 30 and 1000. Groups are representing the amount of performance support category. These values depend strongly on size and kind of content. Therefore, those values might be adapted to provide better performance.

```
# former ttkf.server.search.max.group.count
ttkf.server.search.epss.max.group.count=50
ttkf.server.search.epss.group.limit= 1000
```

Size of search result

By ignoring stored values for several fields, the size of the returned search result is significantly decreased. This is done by using a so-called DocTransformer, an object which manipulates a search result on solr side. The DocTransformer is located in the */lib* directory of the solr core, which is provided by tts. Configuration of the DocTransformer is made in corresponding *solrconfig.xml*.

Index optimization

Optimizing the solr index is an operation, which should not be done very often, since it requires much time and space doing this. That's why index optimization will be triggered only after reindexing and through a new internal job, with configurable interval.

New parameter to configure an internal job to optimize the solr search index. Value is given in hours; default value is 24 hours.

```
ttkf.server.search.optimize.solr.index.interval = 24
```

5.8 Updating from version 2018 to 2018 R2

Solr

In 2018 R2, the supported Solr version is 7.4.0.

This version of Solr is not run as a webapp (meaning it is not a `.war` file) anymore but is run as a service on its own or as a Docker container.

For the setup of Solr as a service or Docker container, we refer to the installation manual.



Make sure that Solr is running before the Tomcat webserver is starting up.

5.9 Updating from version 2017 R2 to 2018

Remove testcases and test result in the DB

In tts performance suite 2018, the Validator has been removed completely.

If there are still testcases and test results in the DB, it is important that the PS consultant deletes these documents via the older tts performance suite instance BEFORE migrating to version 2018.

These documents can be identified by selecting the corresponding content type in the search options.

After the update to 2018, this will not be possible anymore!

In case the update is performed, and the test cases and test results are NOT deleted, there will be errors with structured entities that have associations to testcases or test results.

ALL documents of this entity won't be accessible, and the following error message appears as an exception cause in the server log:

```
Caused by: java.lang.IllegalArgumentException: No enum constant
de.tts.bd.model.content.document.DocumentContentType.Test
```

This error can only be solved by starting an older tts performance suite version with this DB and deleting the testcases and test results as described above.

Websocket support

In tts performance suite 2018, the modern technology called Websockets is now used. To ensure this technique runs properly, servlet filters with mapping `/*` in application server Apache Tomcat must configure the attribute `async-supported` with value `true`.

Application-wide, tts is handling this configuration in the application's `web.xml`. On the application server level, there might be servlet filters configured in the base `web.xml` as well. In this case, these entries must set the attribute as long as the mapping concerns the Curator.

Example:

```

<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>
org.springframework.web.filter.CharacterEncodingFilter
  </filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
  <init-param>
    <param-name>forceEncoding</param-name>
    <param-value>>true</param-value>
  </init-param>
  <async-supported>true</async-supported>
</filter>

<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

```

Troubleshooting

When facing following lines in an exception, then there are still filters missing the async-supported attribute:

Async support must be enabled on a servlet and for all filters involved in async request processing. This is done in Java code using the Servlet API or by adding "true" to servlet and filter declarations in web.xml. Also, you must use a Servlet 3.0+ container



When using AJP, please read the chapter **Configure websocket support for Apache Webserver with AJP** in the appendix.

Migrating from SimpleLDAP login module to new LDAP login module

When changing from the SimpleLDAP login module to the new LDAP login module (users persisted in the database and authorized via LDAP) the existing local users need to be reimported with a changed attribute.

Exporting the users

To export the users to a Microsoft Excel sheet please follow the tts performance suite manual.

Changing the 'Authorization mode' attribute

After exporting the users open the Microsoft Excel sheet and change the 'Authorization mode' attribute for the users that should have local persisted profile but that should authorize via LDAP: (the column may differ if using a customized settings XML file)

Username (login)	Authorization mode
MüllerMa	local
SchmitzSt	local
MaierMi	local
BayerBe	local

Username (login)	Authorization mode
MüllerMa	ldap
SchmitzSt	ldap
MaierMi	ldap
BayerBe	ldap

After changing the attributes, re-import the users by following the Administrator manual.

Changing your config to use the new login module

In the application configuration file, the login module must be set to

```
ttkf.server.user.loginModules=
com.tts.serverfoundation.authentication.ldap.LdapLoginModule
```

For the use case described above, no populator needs to be set as a new populator, the *LdapAuthoritiesPopulator*, is used as its the default populator for the new login module.

The other parameters describing connection parameters and ldap specific settings have not changed and still need to be set.

Migrating external users to local users for better workflow and collaboration

For better collaboration the mechanism for owners and assignees has been refactored. Instead of showing either all external users or none of them, now the external users are filtered based on their persisted permissions. The requirement for this feature is to import all authors from the external identity provider (e.g., ldap), like described above.

If this way of filtering authors is not desired, the previous way can be restored by configuring this parameter:

```
ttkf.server.userlist.workflowcheck=false
```

Configuration change when WebAccess/Curator is embedded in another LMS

To prevent clickjacking inside of our webapps, we prevent that our webapps (WebAccess/Curator) are embedded in other webpages. This means, as default behavior, a LMS can't include the WebAccess in an iframe!

To allow this usecase, you have to define a whitelist of allowed domains which can embed our webapp:

```
ttkf.server.content.security.policy.frame.ancestors=<Comma separated list of domains>
```

The Internet Explorer supports only an old header, which is restricted to one domain. This means that if a list of domains is maintained, the Internet Explorer allows only the first domain. All other browsers allow the whole list of domains.

CSRF-Tokens enabled per default

Starting with release 18.0 *ttkf.server.properties.validateCSRF* will default to "true" instead of "false". When CSRF validation is activated, POST, PUT and DELETE requests must contain a valid CSRF token.

Customizing: Add the following tag to your jsp to add CSRF tokens to forms and ajax calls automatically.

```
<tts:callTemplate name="globalCSRFJavaScriptJSP">
```

JSESSIONID is no longer accessible via URL

JSESSIONID is no longer accessible via URL. Reason: This was a security vulnerability (sensitive information in URL)

Customizing adjustment

QA loading icon in initial_2013

In version 2018 (initial_2013 customizing) a loading icon can be displayed in the QuickAccess when a search request is taking too long.

Please consider the following points when you are changing a customized WebAccess. There is a new template in the *templates.xml*:

The file *config\initial_2013\templates\epss\epss.js.jsp* has changed and now contains the new function *showSearchLoadingIcon*.

The file *config\initial_2013\skin\epss\epss.css* has changed, too, and contains the CSS configuration of the loading icon.

The *QuickAccessSearchLoadingIcon* template has to be called in the following templates:

```
config/initial_2013/templates/epss/epss.no.access.body.jsp
config/initial_2013/templates/epss/epss.no.connection.body.jsp
config/initial_2013/templates/epss/epss.not.found.body.jsp
config/initial_2013/templates/epss/epss.result.body.jsp
```

CSRF token

See above.

XSS fixes:

```
global.jspf
search.form.jsp
search.form.extended.js.jsp
search.form.extended.jsp
search.results.view.header.entry.jsp
deeplink.copy.clipboard.jsp
document.link.jsp
epss.result.link.jsp
epss.qa.docu
ment.link.jsp
epss.qa.js.jsp
epss.qa.jsp
guide.importer.confirm.jsp
guide.list.item.import.jsp
```

See new¬eworthy - 2018 -> Changes in JSP (to prevent XSS) for detailed information.

5.10 Updating from version 2015 R2 to 2016 R2

5.10.1 Eps and Performance Support Categories

Update and configuration

In version 2016 R2 the Performance Support Categories were introduced. Per default the results in QuickAccess will now be grouped by the performance support category model. To prevent this and to keep the former behavior of QuickAccess, the following configuration parameter must be set.

```
ttkf.server.epss.mode=class
```

5.11 Updating from version 2014 R2 to 2015 R2

5.11.1 Password Management for more Security

In 2015 R2 a new password management was introduced. Now the administrator is able to reset passwords and define security rules for passwords like length, strength and history.

These new configuration parameters can be used to define the password security.

Name	ttkf.server.security.password.reset.active
Description	This is the global option to enable (true) or disable (false) the resetting of passwords, both initial passwords and expired ones.
Possible values	true false
Default	false

Name	ttkf.server.security.password.history.size
Description	Specifies the number of passwords in the history. A new user password must be different from the ones contained in the history. Each user has their own history.
Possible values	Positive integer
Default	4

The server provides restrictions for new passwords.

The following restrictions exist:

- Minimum character length (default 7)
- Username may not be contained in password
- First and last name may not be contained in password.
- RegEx condition. Default settings are:
 - At least one lower case character.
 - At least one upper case character.
 - At least one number.
 - At least one special character.

By default, 3 of the 4 conditions must be met.

- The last N passwords of a user cannot be re-used as new password.

The following optional settings exist to configure the password restrictions:

Name	ttkf.server.security.password.min.length
Description	Minimum character length for a new password.
Possible values	Positive integer
Default	7

Name	ttkf.server.security.password.regex.condition.enable
Description	<p>Activate the regex conditions for new passwords. The conditions are:</p> <ul style="list-style-type: none"> • At least one lower case character. • At least one upper case character. • At least one number. • At least one special character.
Possible values	true false
Default	true

Name	ttkf.server.security.password.regex.condition.min.matches.to.pass
Description	At least n regex conditions must match for a new password (see the previous setting for more information about regex conditions).
Possible values	Integer between 0 and 4
Default	3

Name	ttkf.server.security.password.expiration.days
Description	Specifies for how long (days) a password is valid. -1 disables the expiration check.
Possible values	Positive integer up to 365 or -1
Default	90

5.12 Updating from version 2013 R2 to 2014

5.12.1 Solr Search Engine

With the 2014 update, the Solr search server replaced the lucene search engine.

For a general Solr Installation Guide, see *Solr Installation* in the *Preparations* chapter.

On how the Solr server is linked with the tts server components, see the *Solr installation example* in the *Installation step-by-step* chapter.

5.12.2 New mandatory application properties


Name	ttkf.server.search.server.url
Description	This mandatory parameter defines how the Curator and WebAccess can reach the Solr server.
Required	true
Possible values	Full context path to the solr instance
Example	http://127.0.0.1:8983/solr/core

Name	ttkf.server.properties.internal_acceleratorURL
Description	The parameter <i>internal_acceleratorURL</i> is now mandatory, as only the Curator can re/index entities with Solr, the Curator has to know the WebAccess URL.
Required	true for Curator, false for WebAccess
Possible values	Full URL of the WebAccess
Example	http://192.168.86.44:9980/webaccess

Name	ttkf.server.guidabase.rootdir
Description	To store Guides, a Guide-Repository is needed. Therefore, a path is needed, where the Guides can be stored.
Required	true
Possible values	Path with read/write access
Example	C:/TTKF/repository/guide

5.12.3 New optional application properties

Name	ttkf.server.search.max.result.size
Description	This optional parameter defines the maximum number of unfiltered search results which should be returned by a query.
Required	false
Possible values	Positive integer
Default	10.000

Name	ttkf.server.search.disjunction.tie
Description	<p>This optional parameter specifies the tie breaker for the DisMax parser in the SolR search engine.</p> <p>When searching multiple fields, this factor defines the impact that additional matches will have on the score of a result item. For example, when a query matches the title and the description of a document, with the best match found in the title, the score of the additional match in the description will be multiplied by the tie factor and added to the score of the best match in the title of the document.</p> <p>With a tie factor of 0.0, additional matches will be ignored.</p> <p>With a tie factor of 1.0, the overall score of a result item will be the sum of all field matches.</p> <p>Usually, the tie factor should be much less than 1.0.</p> <p> See http://docs.lucidworks.com/display/solr/The+DisMax+Query+Parser for more details.</p>
Required	false
Possible values	Number between 0 and 1
Default	0.5

Search service (Highlighting parameter)

The following configuration parameters for the highlighting of query results are optional. Each highlighting parameter has a direct equivalent in the SolR syntax.



See: <http://docs.lucidworks.com/display/solr/Highlighting> for more details.

Name	ttkf.server.search.highlighting.snippets
Description	Maximum number of highlighted snippets per field.
Required	false
Possible values	Positive integer
Default	1

Name	ttkf.server.search.highlighting. fragsize
Description	Size in characters for highlighted fragments.
Required	false
Possible values	Positive integer
Default	256

Name	ttkf.server.search.highlighting.simple.pre
Description	HTML markup to insert at the start of a match in the fragment.
Required	false
Possible values	HTML markup
Default	
Example	<p>

Name	ttkf.server.search.highlighting.simple.post
Description	HTML markup to insert at the end of a match in the fragment.
Required	false
Possible values	HTML markup
Default	
Example	</p>

Name	ttkf.server.guidebase.paging.size
Description	When the user scrolls in a category and reach the end of the list of currently visible Guides, the next {ttkf.server.guidebase.paging.size} Guides will loaded.
Required	false
Possible values	Positive integer larger than 20
Default	20

LDAP

To consider nested groups for LDAP authentication define the following new optional parameters:

Name	ttkf.server.user.ldap.default.recurseIntoGroups
Description	Should parent groups be considered?
Required	false
Possible values	true false
Default	false

Name	ttkf.server.user.ldap.default.group.base.dn
Description	Base DN for group search.
Required	false
Possible values	root domain for groups
Default	Value of parameter <i>ttkf.server.user.ldap.default.base.dn</i>
Example	DC=teamtraining,DC=local

Name	ttkf.server.user.ldap.default.dialect.group.filter
Description	Filter query for group search. Template parameters can be used.
Required	false
Default	(&(objectClass=group)(dn=\${ttc:group}))

Name	ttkf.server.user.ldap.default.dialect.group.attr.groupMembership
Description	Name of the group membership attribute.
Required	false
Default	memberOf

5.13 Updating from version 2013 to 2013 R2

5.13.1 Curator supports additional workflow functions

- In 2013 R2, two new workflow functions were introduced to offer more flexibility within the workflow process: *NotifyAssigneeOnlyFunction* and *NotifyOwnerOnlyFunction*.
 - *NotifyOwnerOnlyFunction*: Sends an e-mail to the document's owner when the workflow status changes
 - *NotifyAssigneeOnlyFunction*: Sends an e-mail to the document's assignee when the workflow status changes

5.14 Updating from version 2012 R2 to 2013

5.14.1 WebAccess

WebAccess with new design (hasFrameset)

- If using the new design for the WebAccess, a new parameter must be configured within the file *templates.xml*, named *hasFrameset*.

```
<property name="hasFrameset">true</property>
```

- WebAccess designs prior 2013 need to set this value to true (default value), new designs must set this value to false, since the new design has no frames anymore.

WebAccess offers image formats for user profiles

- With tts performance suite 2013, a new design for the WebAccess has been introduced, allowing users to upload profile images. Those images may be used for comments, newsfeeds, or other resources.
To ensure a performant way of accessing content, you are advised to define image formats, so a properly scaled image exists for each use case, reducing network traffic and CPU usage.
- Currently, we provide formats for
 - Profile images
 - Images for newsfeed
- Configuration is done in the properties service following a specific syntax
 - constant value: `ttkf.server.properties.image.format`
 - domain: either `userprofile` or `newsfeedentry`
 - format name: any name
 - value: width and height of the format in pixels; if both values are specified, the image will be scaled to match the longer side; if only one value is given, the other one must be marked with "~"

Examples:

```
ttkf.server.properties.image.format.userprofile.passport=
width:50;height:150
ttkf.server.properties.image.format.newsfeedentry.resource=
width:500;height:~
```

- Configure the productive image formats – comma-separated – via following parameter:

```
ttkf.server.properties.image.formats=
userprofile.passport,newsfeedentry.resource
```

- To limit the file size for profile images, use the following parameter:
 - constant value: `ttkf.server.properties.image.upload`
 - domain: *userprofile*
 - *maxsize*
 - value: any decimal number followed by unit *kb* or *mb*

Examples:

```
ttkf.server.properties.image.upload.userprofile.maxsize=128kb
ttkf.server.properties.image.upload.userprofile.maxsize=1.4mb
```

5.14.2 WebAccess with more flexible Windows-based SSO configuration

Now it is possible to configure the authorities populator for the Windows-based SSO in a more flexible way. Instead of requesting user information and authorities from an LDAP server (default behavior), you may now get those bits of information from the local database.

Example for configuring local database authorities populator:

```
ttn.bd.login.sso.windows.authorization.WindowsSSODatabaseAuthorities
Populator
```

For more information, see *User service for authentication and authorization* in the appendix.

5.14.3 Curator & WebAccess: Combine attributes in LDAP groupMembership

The definition of the groupMembership attribute now allows the combination of different attributes from LDAP.

Example:

```
ttkf.server.user.ldap.default.dialect.user.attr.groupMembership=#c#l
```

With the value *c* standing for country and *l* for location, the LDAP-interface now creates entries like "c=DE,l=Heidelberg".

5.14.4 Curator supports definition of more mime-types for zip archives

- When uploading a zip file as document content, the archive is extracted and its content examined.

Files that match a set of pre-configured mime types will be considered to be different publishing formats of the document and will be linked in the WebAccess automatically.

The files considered as published document formats are configured in a pair of mime type / priority values.



A duplicate definition of a priority value will overwrite the previously defined mime type value.



When extending the list of supported mime types with custom mime types, make sure to respect priority values that are already used. This means that you should use incremental values starting with the value 10.

Preconfigured default values

The following values are preconfigured and therefore do not require explicit definition:

- text/html=0
- text/xml=1
- text/rtf=2
- application/msword=3
- application/vnd.ms-word=4
- application/mspowerpoint=5
- application/vnd.ms-powerpoint=6
- application/msexcel=7
- application/vnd.ms-excel=8
- application/pdf=9
- Definition of custom mime-type support
 - The list of supported mime types can be extended using the *application-config.properties* file. The mime types have to be added in the format {prefix}.{mimetype}={priority} where the prefix is

ttkf.integrator.collectionContentHandler.customSupportedTypes.

The following example shows how to add support for javascript (*.js), richtext (*.rtx) and plain text (*.txt) files:

```
ttkf.integrator.collectionContentHandler.customSupportedTypes.text/javascript=10
ttkf.integrator.collectionContentHandler.customSupportedTypes.text/richtext=11
ttkf.integrator.collectionContentHandler.customSupportedTypes.text/plain=12
```

Adding the following lines will additionally add support for MS Help files (*.chm)

```
ttkf.integrator.collectionContentHandler.customSupportedTypes.application/mshelp=13
ttkf.integrator.collectionContentHandler.customSupportedTypes.application/vnd.ms-htmlhelp=14
```

5.15 Updating from version 2012 to 2012 R2

5.15.1 Configuration of the WebAccess

template.xml

- The *template.xml* contains an additional filter configuration.

The *roleAccessPredicateFilter* is now part of all filter chains by default.

This filter suppresses all entities that are not assigned to any role the current user is a member of.

In earlier releases, the *roleAccessPredicateFilter* was applied implicitly to all filter chains. Therefore, to maintain the default behavior, it is necessary to add the filter to all existing filter chains. Take care to add it to existing customizings if it is needed. (it will be needed in most cases.)

Example:

```
<filter-chain name="coursetreeChain">
  ...
  <add-filter name="roleAccessPredicateFilter" passonPreview="true">
  ...
</filter-chain>
```

- The *template.xml* contains an additional attribute in the filter chain configuration. All defined filter chains (including the nameless default filter chain) have to define an additional mandatory *chain-elements-factory* attribute. The only valid value for that attribute, at the current state, is *de.tts.bd.eud.filter.chain.DefaultChainElementFactory*.

Example:

```
<filter-chain name="coursetreeChain"
chain-elements-
factory="de.tts.bd.eud.filter.chain.DefaultChainElementFactory">
  ...
</filter-chain>
```

feature.xml

Until this version, the path of the *feature.xml* file could only be a path inside the application classpath. From now on, the path can also be an absolute path anywhere on the system. The path has just to begin with the prefix "file:"

5.16 Updating from version 7.1 to 2012

- The content-based revision number, which could be combined with a storage pattern, has been removed. Please use the document type assigned versioning instead, which provides a lot more possibilities.
- The "showStatifyCertificateButton" parameter was added. For more details, see the *Pro- perties service* chapter.
- The default maintenance interval of the repository service was decreased to 5 seconds.
- Detailed logging of LDAP connections has been introduced. See the *Logging service* chapter for more information.

Appendix

6.1 Properties service

Curator & WebAccess

Application-wide settings are defined in the properties service.

Name	acceleratorContextPath (Curator only)
Description	Defines the site relative URL to the WebAccesss. In case several WebAccesses are installed, the one used for preview should be given here.
Possible value	/webaccess/

Name	integratorContextPath (WebAccess only)
Description	Defines the site relative URL to the Curator.
Possible value	/curator/

Name	superuserIPs
Description	A comma-separated list of IP addresses for which super user functions, like database initialization, are permitted. Do make sure that this property is not left blank (read: you should at least enter "127.0.0.1" here.)
Possible value	127.0.0.1

Name	allowLoginCookie
Description	Allows remembering login data by means of a browser cookie.
Possible value	yes (default) no

Name	automanageDocumentLanguage
Description	Automatically determines in which language multilingual meta data are saved (by using content language).
Possible value	yes (default) no

Name	xmlReaderClass
Description	Defines an XML reader if the application server does not supply one. (This property is removed since 2021 R2)
Possible value	Fully qualified Java class name. Example: <i>org.apache.xerces.parsers.SAXParser</i>

Name	applicationServerLogDirectory
Description	Defines a directory path for application server log files. If set, the administrative function download log files include the log files from the server.
Possible value	Any valid path on a file system, e.g., <i>c:/myAppServerDir/logs</i> .

Name	checkedOutLicenseValidity
Description	Sets the number of days a checked-out license remains valid.
Possible value	Any positive integer value, e.g., 14.

Name	httpCacheValidation
Description	Defines the method of HTTP cache validation.
Possible value	ETag LastModified (default)

Name	httpCacheExpiration
Description	Defines the method of HTTP cache expiration.
Possible value	Expires CacheControl (default)

Name	httpCacheExpirationAge
Description	Defines the age of HTTP cache expiration sent by the server in seconds.
Example	31556926 (1 year) (default)

Name	showStatifyCertificateButton
Description	If set to true, the option "Show Certificate Button" within the stratification is enabled.
Possible value	true (default) false

Name	qaIndexLength
Description	Sets the maximum length of an index entry in the QuickAccess signature.
Possible value	Any positive integer value, e.g., 64. 0 = no restriction

Name	qaWildcardRestriction
Description	Restricts level-wildcard assignments (generic signature).
Example	GEN:1001;URL:1001;SAP:10001 1 = level is required and cannot be used as a wildcard 0 = level can be assigned as a wildcard by the user

Name	serverUrl
Description	Represents the URL and context path of the application used to backtrack imported objects to their origin.
Possible value	Any valid URL. Example: http://sampleServer:8080/curator

Name	external_integratorURL
Description	Represents the URL to access the Curator from external locations, for example in a reverse proxy scenario. External URL and the address for login usually are the same. Therefore, the URL should be equal concerning case sensitivity, otherwise firewalls or proxies might block these URLs. If not specified the external URL of the Curator is mapped to the internal URL.
Possible value	Any valid URL. Example: http://externalServer/curator
Attention	This parameter is required for creating user generated content using the Creator.

Name	external_acceleratorURL
Description	Represents the URL to access the WebAccess from external locations, for example in a reverse proxy scenario. External URL and the address for login usually are the same. Therefore, the URL should be equal concerning case sensitivity, otherwise firewalls or proxies might block these URLs. If not specified the external URL of the WebAccess is mapped to the internal URL.
Possible value	Any valid URL. Example: http://externalServer/webaccess

Name	internal_integratorURL
Description	Represents the URL to access the Curator in an internal way, mainly used for internal server communication between the Curator and WebAccess. This property is mandatory!
Possible value	Any valid URL. Example: http://internalServer:8080/curator

Name	internal_acceleratorURL
Description	Represents the URL to access the WebAccess internally, mainly for internal server communication between the Curator and WebAccess. This property is mandatory!
Possible value	Any valid URL. Example: http://internalServer:8080/webaccess

Name	portalLoginRequired (WebAccess only)
Description	Defines whether WebAccess requires login or not.
Possible value	true (default) false

Name	portalIgnoresMaintenanceRoles (WebAccess only)
Description	Defines whether the maintenance roles are ignored or not when it comes to filtering objects visible to a user.
Possible value	true false (default)

Name	portallgnoresMaintenanceRolesOnPreview (WebAccess only)
Description	Defines whether the maintenance roles are ignored on preview or not when it comes to filtering objects visible to a user.
Possible value	true false (default)

Name	qaSearchMode (WebAccess only)
Description	Defines the QuickAccess search mode.
Possible value	"direct" searches only documents of the context specified by the QuickAccess application. "cascade" uses multiple searches to allow more general results. E.g., if the specific context is "GEN;excel;cell formatting", the cascade mode will also search for the context "GEN;excel" if no documents were found for the specified context. "all" uses multiple searches defined by searchlists "qaDirectSearchList" and "qaIndirectSearchList" (default). "off" disables document search.

Name	qaGlossarySearchMode (WebAccess only)
Description	Enables QuickAccess to search within glossary entries.
Possible value	on (default) off

Name	qaDirectSearchList (WebAccess only)
Description	Sets a comma separated list of signature-level maps for a direct result list.
Example	GEN:1110;URL:1110;SAP:11000 "1110" means the first three levels of a document's signature must be equal to the signature currently searched for.

Name	qaIndirectSearchList (WebAccess only)
Description	A comma-separated list of signature-level maps for an indirect result list.
Example	GEN:1100;URL:1100;SAP:00000

Name	qaSortResult (WebAccess only)
Description	Sets sorting of the result list by explorative context.
Possible value	yes (default) no

Name	qaSortAdjacentResult (WebAccess only)
Description	Sets sorting of the adjacent result list by explorative content.
Possible value	yes no (default)

Name	qaFilterChain (WebAccess only)
Description	Sets the filter chain which filters the documents returned by the QuickAccess. If none is specified, the default filter chain is used.
Example	qaFilterChain

Name	accessLatestReleased (WebAccess only)
Description	Defines whether the latest released document version is displayed (=true), or the latest version of a document (=false) (without workflow engine).
Possible value	true false (default)

Name	image.format.\${domain}.\${name}
Description	Defines an image format by providing the <i>domain</i> and <i>name</i> of the format. Currently provided domains are <i>userprofile</i> and <i>newsfeedentry</i> . The value specifies width first, and height last. If given both values, the image is scaled along the greater side; if given one value, the other one must be marked with "~".
Example	image.format.userprofile.passport = width:50;height:150 image.format.newsfeedentry.default = width:500;height:~

Name	image.upload.\${domain}.maxsize
Description	<p>Defines the max size an image for the providing domain may have to get uploaded successfully. Currently, the only supported domain for this parameter is userprofile. The value is given as a decimal number directly followed by the unit kb or mb.</p> <p>Default value: 1.6mb</p>
Example	<pre>image.upload.userprofile.maxsize = 128kb image.upload.userprofile.maxsize = 1.44mb</pre>

Name	image.formats
Description	Defines the list of productive image formats as comma-separated list.
Example	<code>image.formats = userprofile.passport,newsfeedentry.resource</code>

Name	portalLinkBaseURL
Description	Defines the base URL which shall be used for the portal link. This parameter is optional.
Example	<code>http://server:port/context</code>

Name	ttkf.server.properties.reject.user.agents.patterns
Description	<p>This parameter is optional.</p> <p>In chromium-based browsers deeplinks are opened in two tabs when clicked within Microsoft Office products, such as Word or Excel. Reason for this behavior is an internal request with an internal http Producer to check the link itself. With the following parameter it is possible, to reject requests made by an user agent containing the given comma-separated patterns. The strings are case-insensitive.</p>
Example	<code>ttkf.server.properties.reject.user.agents.patterns = microsoft,ms-office</code>

Name	ttkf.server.contentConfig.deactivated
Description	By setting the parameter <code>ttkf.server.contentConfig.deactivated</code> to a comma separated list of configuration names, these configurations can be deactivated. They cannot be selected for creating new documents anymore. However, playing content created with deactivated configs is still possible as

	well as editing and republishing these documents.
Example	<code>ttkf.server.contentConfig.deactivated=initial,my_custom_config,another_config</code>

6.2 Data service

For tts server to be able to store data persistently, a connection to the previously created database is necessary. The Hibernate ORM framework is used to get object-oriented access to the relational database. Therefore, all defined properties are delegated to Hibernate and the connection pooling framework.

6.2.1 JNDI data source

We strongly recommend you configure the database connection by creating the data source in the application server. A data source is a logical database interface, providing (and hiding complex) information with regard to the database connection and connection pooling. Each data source can be accessed through a JNDI name.

Database configuration

If a database connection is provided through a JNDI data source, the JNDI name, the Hibernate dialect, and the name of the database schema have to be specified. Additional information, like the JDBC connection URL and the database user, are provided by the JNDI data source.

Service name: data (maintenanceInterval: 30)

Name	hibernate.dialect
Description	Defines the fully qualified Java class of the database dialect (type of database and used character set).
Possible value	Example: <i>de.tts.bd.business.data.UnicodeOracle12cDialect</i> For a complete list, please see the supported Hibernate dialects below.

Name	hibernate.default_schema
Description	Sets the name of the default schema to use for SQL statements (usually the name of the database user).
Example	TTS

Name	portal.hibernate.dialect
Description	Defines the fully qualified Java class of the database dialect (type of database and used character set) for tts Server WebAccess.
Possible value	Example: <i>de.tts.bd.business.data.UnicodeOracle12cDialect</i> For a complete list, please see the supported Hibernate dialects below.

Name	portal.hibernate.default_schema
Description	Sets the name of the default schema to use for SQL statements (usually the name of the database user) for TTS Server WebAccess.
Example	TTS



For Oracle, the default schema should always be specified in upper case.

The necessary parameter can be set via the *application-config.properties* file, for example:

```
# curator database configuration
ttkf.server.data.hibernate.dialect =
de.tts.bd.business.data.UnicodeOracle12cDialect
ttkf.server.data.hibernate.default_schema = TTS
```



The Hibernate dialect ensures that the suitable syntax is used for initializing, updating, and accessing the database. Any later change of the dialect (for example from non-Unicode to Unicode) may cause problems.

Supported Hibernate Dialects

```
de.tts.bd.business.data.UnicodeSQLServerDialect
de.tts.bd.business.data.UnicodeOracle12cDialect (suits 18c as well)
de.tts.bd.business.data.Oracle12cDialect
de.tts.bd.business.data.SQLServerDialect
```



The JDBC driver for Microsoft SQL Server requires the property `select Method` to be set to the value `cursor`. If there are performance issues, the value could be changed to `direct`.

6.3 Store service

The store service manages the runtime files used by the TTKF Server. The only property that needs to be set is the base path where the files will be stored. Within this directory, all sub-directories that are necessary for the single stores (config, document pool, etc.) will be created automatically.

In case you want to have separate stores (for pool, image, config), you can define them independent from the base path.



Temporary stores, as used by the report and import/export engine, will be located in a sub-directory of the data store base path. These stores cannot be defined separately from the data store.



Please make sure the data store provides enough disk space for storing temporary runtime files. See the requirements section for details.



For stores which are located under Minlo, s3 url are needed. For stores on a file system, a file url has to be provided

Service name: store

Name	store.base.directory
Description	The base directory of the data store.
Example	<i>/home/ttn/datastore</i>

Name	store.spool.url <optional>
Description	Defines the spool directory where document contents are stored temporarily before being uploaded to the repository.
Example	<i><u>file:///home/ttn/spool</u></i> <i><u>s3://bucket/ttn/spool</u></i>

Name	store.config.url <optional>
Description	Defines the directory where the version-controlled configurations of the tts performance suite Server Producer are stored.
Example	<i><u>file:///home/ttn/config</u></i> <i><u>s3://bucket/ttn/config</u></i>

Name	store.index.directory <optional> <deprecated>
Description	Defines the path where the search engine stores its index files.
Example	This is no longer used. Please Remove.

Name	store.image.url <optional>
Description	Defines the directory where images uploaded through the Curator are stored.
Example	<i><u>file:///home/ttn/image</u></i> <i><u>s3://bucket/ttn/image</u></i>

Name	store.cre.url <optional>
Description	Defines the directory where content runtime environments for the tts performance suite Desktop Producer is stored by the TTKF Server.
Example	<u>file:///home/ttn/cre</u> <u>s3://bucket/ttn/cre</u>

Name	store.report.url <optional>
Description	Defines the directory where report generated by the tts performance suite Server are stored for download.
Example	file:///home/ttn/report



The report store cannot be an S3 path

6.4 User service for authentication and authorization

Service name: user

Name	burst.retry.size
Description	Sets the maximum permissible number of failed login attempts by a user before the system locks the user name for a specified time.
Possible value	Any integer value, e.g., 2

Name	burst.denial.time
Description	Defines the time (in seconds) a username for which the maximum number of failed login attempts was exceeded will be locked from the system.
Possible value	Any integer value, e.g., 30

6.4.1 Login modules

Login modules are responsible to authenticate a user and provide user-specific information (user profile) to the application. They are categorized primarily by the point in time they are active during the authentication process. Modules which allow for an *explicit authentication* require a username and a password as their input. They are activated as soon as these bits of information have been provided. Modules that support *implicit authentication* derive their information from the first HTTP request to the server. These modules are primarily used for Single-Sign-On.

Name	loginModules
Description	A comma-separated list of fully qualified Java class names of login modules.
Possible value	one of the Login modules listed below
Example	de.tts.bd.business.login.ldap.LdapLoginModule



When using the CookieLoginModule in combination with external users the following parameter has to be configured. Otherwise, the user object cannot be provided.

It is possible to support multiple login systems in the backend with cookie login. Mixed scenarios like database and external authentication are fully featured via cookie authentication. This is done by configuring multiple authorities populators in a comma-separated list. Per default the database authorities populator is always appended to the list of populators.

Name	loginModules.CookieLoginModule.populator
Description	Populators which are used to provide the user object.
Possible value	de.tts.bd.business.login.ldap.ExtendedLdapAuthoritiesPopulator (deprecated), de.tts.bd.business.login.request.RequestElementsAuthoritiesPopulator, ttn.bd.login.sso.windows.authorization.WindowsSSOAuthoritiesPopulator, com.tts.serverfoundation.authentication.saml.SAMLAuthoritiesPopulator

List of Login Modules

	Authentication	Remarks
de.tts.bd.business.login.DefaultLoginModule (explicit)	Authentication with user name + password against data stored in a local database.	
de.tts.bd.business.login.CookieLoginModule (implicit)	Login via user name and login-token contained in an HTTP cookie.	

<p>com.tts.serverfoundation.authentication.LdapLoginModule (implicit)</p>	<p>Authenticates against an LDAP directory.</p>	<p>When User with domain 'ldap' exists in database this user is logged in. Otherwise, the user is created at runtime by obtaining the profile from information stored in an LDAP record on the directory server.</p>
<p>de.tts.bd.business.login.ldap.SimpleLdapLoginModule (explicit) (deprecated)</p>	<p>Authenticates against an LDAP directory.</p>	<p>The user must also exist in the local database. Only the password may be omitted.</p>
<p>de.tts.bd.business.login.ldap.ExtendedLdapLoginModule (explicit) (deprecated)</p>	<p>Authenticates against an LDAP directory.</p>	<p>The user is created at runtime by obtaining the profile from information stored in an LDAP record on the directory server.</p>
<p>de.tts.bd.business.login.request.RequestElementsLoginModule (implicit)</p>	<p>Authenticates against an authentication value stored in the HTTP header or as a URL parameter.</p>	<p>The user is created at runtime by obtaining the profile from information stored in the HTTP request.</p>
<p>ttn.bd.login.sso.windows.WindowsSSOLoginModule (implicit)</p>	<p>Authenticates and authorizes against LDAP with Windows logon credentials (default). Since 2012 R2, it is possible to configure a different authorities populator.</p>	<p>The user is created at runtime by obtaining the profile from information stored in Windows principal object and LDAP record. Since 2012 R2, the user information can be retrieved from the local data- base as well using users with domain 'ldap'.</p>
<p>com.tts.serverfoundation.authentication.saml.SAMLAuthenticationLoginModule</p>	<p>This login module is designed for Single-Sign-On based on SAML v2. The user authenticates against the configured identity provider.</p>	<p>When User with domain 'saml' exists in database this user is logged in. Otherwise, the user is created at runtime by obtaining the profile from information stored in the identity provider.</p>



Login modules may be combined to form module chains. As soon as one of the modules successfully authenticates the user, the user will be logged in.

6.4.2 LDAP authentication

Supported scenarios

For LDAP-related login modules three scenarios are supported.

(A) Local user + LDAP authentication + without service user (deprecated)

This scenario is provided by the *SimpleLdapLoginModule*. It is recommended for directories with a flat structure combined with local user administration.

1. Load a user from the local database
2. Retrieve LDAP authentication information from the user object to find the user's Distinguished Name (DN) and LDAP server.
3. In case no LDAP authentication information is available, the DN is created through a template.
4. Bind the determined DN to the LDAP directory for authentication.

(B) Local user + LDAP authentication + with service user

This scenario is provided by the *SimpleLdapLoginModule*. It is recommended for directories with a complex structure combined with local user administration.

1. Load a user from local database (user needs 'ldap' in domain column, achievable with user import)
2. Bind the service user to LDAP directory.
3. Ascertain the DN of the user's LDAP entry through a search operation.
4. Bind the determined DN to the LDAP directory for authentication

(C) Runtime user + LDAP authentication + with service user

This scenario is provided by the *LdapLoginModule*. It is recommended for directories with structures of any complexity and requires no local user management.

1. Ascertain the DN of the user's LDAP entry through a search operation.
2. Create a runtime user profile by using the data stored in the LDAP entry.
3. Bind the determined DN to the LDAP directory for authentication.

Properties for LDAP configuration only

Name	ldap.default.provider.url
Description	Defines the URL of the LDAP server.
Example	ldap://ldapserver.teamtraining.local:389/

Name	ldap.default.base.dn
Description	Sets the base DN used for all LDAP operations in TTS PERFORMANCE SUITE (root DN).
Example	DC=teamtraining,DC=local

Name	ldap.default.manager.dn (scenarios B,C)
Description	Sets the DN of the service user used for all LDAP search operations.
Example	<p><i>CN=ldapadmin,OU=Services,OU=TTS, DC=teamtraining,DC=local</i></p> <p>In case the DN contains a backslash "\", this character must be escaped with "\", other- wise the value of this parameter might be parsed wrong.</p> <p>Example: <i>CN=ldapadmin, \\ admin,OU=Services,OU=TTS, DC=teamtraining,DC=local</i></p>

Name	ldap.default.manager.password (scenarios B,C)
Description	Sets the password of the service user. In case the password contains a backslash "\", this character must be escaped with "\", otherwise the value of this parameter might be parsed wrong. Best way would be to avoid a backslash in the password and to use another special character.

Name	ldap.default.manager.password.x (scenario B,C)
Description	Sets the password of the service user. Other than <i>ldap.default.manager.password</i> , where the password is needed in cleartext, this parameter uses the BASE64 encoded version of the used password. If both parameters are set the encrypted version is used.

Name	ldap.default.dialect
Description	Fully qualified Java class name of dialect for LDAP server.
Possible value	<p><i>de.tts.bd.business.login.ldap.GenericLdapDialect</i> (default dialect for all LDAP services)</p> <p><i>de.tts.bd.business.login.ldap.ActiveDirectoryDialect</i> (Microsoft Active Directory Service)</p>

Name	ldap.default.user.dn.template (SimpleLdapLoginModule)
Description	Defines the template for DN's of user entries. The placeholder <code>\${ttc:user}</code> may be used instead of a username. During the login process, the placeholder is replaced by the actual username.
Possible value	<code>CN=\${ttc:user},OU=User,OU=TTS, DC=teamtraining,DC=local</code>

Name	ldap.default.mapper.userProfile (ExtendedLdapLoginModule)
Description	Fully qualified name of Java class for mapping LDAP attributes to values of the user profile.
Possible value	<code>de.tts.bd.business.login.ldap.GenericLdapUserProfileMapper</code>

Name	ldap.default.mapper.processRoles (ExtendedLdapLoginModule)
Description	Fully qualified name of Java class for mapping LDAP groups of a user to process roles of the TTS PERFORMANCE SUITE server.
Possible value	<code>de.tts.bd.business.login.ldap.GenericLdapProcessRoleMapper</code>

Name	ldap.default.mapper.maintenanceRoles (ExtendedLdapLoginModule)
Description	Fully qualified name of Java class for mapping LDAP groups of a user to maintenance roles of the TTS PERFORMANCE SUITE server.
Possible value	<code>de.tts.bd.business.login.ldap.GenericLdapMaintenanceRoleMapper</code>

Name	ldap.default.recurseIntoGroups
Description	Should parent groups be considered? Default: false <code>ttkf.server.user.ldap.default.recurseIntoGroups={true false}</code>
Example	<code>ttkf.server.user.ldap.default.recurseIntoGroups=true</code>

Name	ldap.default.group.base.dn
Description	Base DN for group search. Default: Value of parameter <code>ttkf.server.user.ldap.default.base.dn</code> <code>ttkf.server.user.ldap.default.group.base.dn={root domain for groups}</code>
Example	<code>ttkf.server.user.ldap.default.group.base.dn = DC=teamtraining,DC=local</code>

Name	ldap.default.dialect.group.filter
Description	Filter query for group search. Template parameters can be used. Default: <code>(&(objectClass=group)(dn=\${ttc:group}))</code>
Example	<code>ttkf.server.user.ldap.default.dialect.group.filter= (&(objectClass=group)(dn=\${ttc:group}))</code>

Name	ldap.default.dialect.group.attr.groupMembership
Description	Name of the group membership attribute. Default: <code>memberOf</code>
Example	<code>ttkf.server.user.ldap.default.dialect.group.attr.groupMembership=memberOf</code>



You can manage the mapping of LDAP groups to TTS PERFORMANCE SUITE roles (either process or author roles) in the *Assign authorizations to external users* dialog located in the administration section (>*Settings* >*Users*).

LDAP dialects

LDAP dialects are used to simplify the mapping of LDAP entries to values of the user profile. Since most LDAP directories are structured in a very particular way, a specific dialect may not be needed in many cases. At the moment, two dialect classes are supported. The *GenericLdapDialect* provides a set of mappings which may be common to a lot of directories and which should be sufficient in most cases. The *ActiveServerDialect* can be used in a Windows Server environment.

All properties must be prefixed with *ldap.default.dialect*, which is omitted in the following list for improved readability.

Name	.user.filter
Description	<p>Defines the LDAP search filter used to find user entries (e.g. "(&(objectClass=user)(cn=\${ttc:user}))").</p> <p>Default value (GenericLdapDialect): (&(objectClass=user)(cn=\${ttc:user}))</p> <p>Default value (ActiveDirectoryDialect): (&(objectClass=user)(userPrincipalName=\${ttc:user}@\${ttc:adsDomain}))</p>

Name	.user.attr.name
Description	<p>Sets the name of the LDAP attribute containing the username.</p> <p>Default value: cn</p>

Name	.user.attr.uniqueid
Description	<p>Sets the name of the LDAP attribute containing a unique identifier.</p> <p>Default value (GenericLdapDialect): <i>distinguishedName</i></p> <p>Default value (ActiveDirectoryDialect): <i>objectGUID</i></p>

Name	.user.attr.uniqueid.encoding
Description	<p>Sets the encoding of the unique id (allowed here are "string" or "uuid_bin", the latter denotes a binary UUID format).</p> <p>Default value (GenericLdapDialect): <i>string</i></p> <p>Default value (ActiveDirectoryDialect): <i>uuid_bin</i></p>

Name	.user.attr.firstname
Description	<p>Defines the name of the LDAP attribute containing the first name.</p> <p>Default value: <i>givenName</i></p>

Name	.user.attr.lastname
Description	<p>Defines the name of the LDAP attribute containing the last name.</p> <p>Default value: <i>sn</i></p>

Name	.user.attr.email
Description	Defines the name of the LDAP attribute containing the email address. Default value: <i>mail</i>

Name	.user.attr.language
Description	Defines the name of the LDAP attribute containing the language. Default value (GenericLdapDialect): <i>preferredLanguage</i> Default value (ActiveDirectoryDialect): <i>c</i>

Name	.user.attr.language.encoding
Description	Sets the encoding of the language (allowed here are "iso639", "iso3166", "rfc1766", the value is converted automatically). Default value (GenericLdapDialect): <i>iso639</i> Default value (ActiveDirectoryDialect): <i>iso3166</i>

Name	.user.attr.editLanguage
Description	Defines the name of the LDAP attribute containing the edit language (if language is not provided). Default value: <i>preferredLanguage</i>

Name	.user.attr.editLanguage.encoding
Description	Sets the encoding of the edit language (allowed here are "iso639", "iso3166", "rfc1766", the value is converted automatically). Default value: <i>iso639</i>

Name	.user.attr.interfaceLanguage
Description	Defines the name of the LDAP attribute containing the interface language (if language is not provided). Default value: <i>preferredLanguage</i>

Name	.user.attr.interfaceLanguage.encoding
Description	Sets the encoding of the interface language (allowed here are "iso639", "iso3166", "rfc1766", the value is converted automatically). Default value: iso639

Name	.user.attr.groupMembership
Description	Sets the name of the LDAP attribute containing the group membership information (must contain distinguished names of the groups). This parameter offers a combination of LDAP-attributes, delimited by "#". Default: memberOf
Example	(with combined attributes): #c#l results into "c=DE,l=Heidelberg"

Name	.adsDomain (ActiveDirectoryDialect only)
Description	Sets the ADS domain name (e.g., teamtraining.local). If it is not specified, the search includes all domains. Only necessary if... <i>user.filter</i> has not been set. Default value: teamtraining.local

Configuration of LDAP over SSL

To ensure a secure communication between the TTS PERFORMANCE SUITE server and the LDAP server, the use of SSL (Secure Sockets Layer) is strongly recommended. Only two additional steps are necessary to enable SSL:

1. The provider-url must match the URL scheme of ldaps, for example
ldaps://secureldap.teamtraining.local:636.
2. The SSL certificate of the secure LDAP server must be stored in a keystore of trusted certificates and provided to the JVM or the application server. TTS PERFORMANCE SUITE server checks the certificate's availability at runtime

You can import the certificate into the JVM using *keytool*:

```
keytool -import -v -trustcacerts -alias ldapserver -file
ldapserver_cert.cer -keystore trust-store.ks -storepass secret
```

Then, start the JVM with the keystore by setting the JVM parameters *javax.net.ssl.trustStore* and *javax.net.ssl.trustStorePassword*:

```
java -Djavax.net.ssl.trustStore=truststore.ks
-Djavax.net.ssl.trustStorePassword=secret
```



If configuration of the JVM parameter is not possible, the keystore of the system user (*\${user.home}/keystore*) or the keystore of the JVM (*\${java.home}/lib/security/cacerts*) may be used instead.

Migration of deprecated LDAP-configuration (6.3.2 and earlier)

Both properties, *ldap.server* and *ldap.user_dn*, are deprecated and not supported anymore (since version 6.4.1). Please adapt your configuration in the following way:

- *SimpleLdapLoginModule* has to be registered
- *ldap.server* must be replaced with *ldap.default.provider.url*
- *ldap.user_dn* must be replaced with *ldap.default.user.dn.template*

Known restrictions of the LDAP interface

- Only "internal" authentication with username and password is supported. Alternative mechanisms in accordance with SASL (RFC 2222) are not possible.
- Start-TLS (RFC-2830) is currently not supported.
- LDAP groups have to be associated to user entries. Users associated with groups are not supported.

6.4.3 Single-Sign-On (SSO)

Request-based Single-Sign-On

Many SSO systems use elements of the HTTP request (headers, parameters) to indicate if a user is signed in or not. With this knowledge in mind, the generic login module already supports a large number of SSO products.

To enable SSO as authentication mode, the login module *de.tts.bd.business.login.request.RequestElementsLoginModule* must be registered in the property *loginModules* of the user service.

Some of the configuration properties of the SSO authentication have to meet a specific syntax to get the values extracted properly from the HTTP request. Below, the extractors for particular elements of the HTTP request are described.

Configuration of LDAP over SSL

To ensure a secure communication between the TTS PERFORMANCE SUITE server and the LDAP server, the use of SSL (Secure Sockets Layer) is strongly recommended. Only two additional steps are necessary to enable SSL:

The provider-url must match the URL scheme of ldaps, for example

```
ldaps://secureldap.teamtraining.local:636.
```

1. The SSL certificate of the secure LDAP server must be stored in a keystore of trusted

certificates and provided to the JVM or the application server. TTS PERFORMANCE SUITE server checks the certificate's availability at runtime

You can import the certificate into the JVM using *keytool*:

```
keytool -import -v -trustcacerts -alias ldapserver -file
ldapserver_cert.cer -keystore truststore.keystore -storepass secret
```

Then, start the JVM with the keystore by setting the JVM parameters *javax.net.ssl.trustStore* and *javax.net.ssl.trustStorePassword*:

```
java -Djavax.net.ssl.trustStore=truststore.keystore
-Djavax.net.ssl.trustStorePassword=secret
```



If configuration of the JVM parameter is not possible, the keystore of the system user (*{user.home}/.keystore*) or the keystore of the JVM (*{java.home}/lib/security/cacerts*) may be used instead.

Migration of deprecated LDAP-configuration (6.3.2 and earlier)

Both properties, *ldap.server* and *ldap.user_dn*, are deprecated and not supported anymore (since version 6.4.1). Please adapt your configuration in the following way:

- SimpleLdapLoginModule has to be registered
- *ldap.server* must be replaced with *ldap.default.provider.url*
- *ldap.user_dn* must be replaced with *ldap.default.user.dn.template*

Known restrictions of the LDAP interface

- Only "internal" authentication with username and password is supported. Alternative mechanisms in accordance with SASL (RFC 2222) are not possible.
- Start-TLS (RFC-2830) is currently not supported.
- LDAP groups have to be associated to user entries. Users associated with groups are not supported.

6.4.4 Single-Sign-On (SSO)

Many SSO systems use elements of the HTTP request (headers, parameters) to indicate if a user is signed in or not. With this knowledge in mind, the generic login module already supports a large number of SSO products.

To enable SSO as authentication mode, the login module *de.tts.bd.business.login.request.RequestElementsLoginModule* must be registered in the property *loginModules* of the user service.

Some of the configuration properties of the SSO authentication have to meet a specific syntax to get the values extracted properly from the HTTP request. Below, the extractors for particular elements of the HTTP request are described.



The value of an extractor contains two elements separated through a dot - the first re-

presenting the type of extractor, the second the value to extract. For example, the configuration ***parameter.user*** creates an extractor which reads the value of the URL parameter *user*.

Extractor	Parameter	Description	Values
Parameter	Name of a request parameter	Extracts a value from a URL parameter	/webaccess/index.do?user=musterman (extractor parameter.user determines the value "musterman")
parameterValues	Name of a request parameter	Extracts a list of values from all URL parameters with the same name	/webaccess/index.do?group=sales&group=accounting (extractor parameterValues.group determines the values {"sales", "accounting"})
Header	Name of a request header	Extracts a value from an HTTP header	GET /webaccess/index.do ... X-User: musterman ... (extractor header.X-User determines the value "musterman")
Cookie	Name of a Cookie	Extracts a value from a Cookie	GET /webaccess/index.do ... Cookie: email=musterman@mustercompany.local; username=musterman ... (extractor cookie.email determines the value "musterman@mustercompany.local")
Constant	Constant value	Simply extracts the value given to the parameter, not from the HTTP request	extractor constant.en-us determines the value "en-us"



The configuration options of the SSO login module are divided between properties concerning the authentication and properties used to map value to the user profile.

Name	request.auth.value
Description	Extracts a value to validate a user. If the value is available and could be validated successfully, the user is authenticated.
Example	header.user (extractors returning a value list as <i>parameterValues</i> are not allowed)

Name	request.auth.validator
Description	Defines the validator which validates the returned value of property <i>request.auth.value</i> .
Example	notEmpty (currently, only the notEmpty validator exists. Thus, a user is authenticated if the value of <i>request.auth.value</i> is not empty)

Name	request.attr.name
Description	Extracts the (login) name of a user.
Example	parameter.user (extractors returning a value list as <i>parameterValues</i> are not allowed)

Name	request.attr.uniqueId
Description	Extracts a unique identifier (e.g., email address or the username again).
Example	parameter.email

Name	request.attr.firstname
Description	Extracts the first name of a user.
Example	parameter.firstname

Name	request.attr.lastname
Description	Extracts the last name of a user.
Example	parameter.lastname

Name	request.attr.email
Description	Extracts the e-mail address of a user.
Example	parameter.email

Name	request.attr.language
Description	Extracts the language of the user which will be used as the edit and the interface language within the TTS PERFORMANCE SUITE server.
Example	constant.en (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	request.attr.language.encoding
Description	Specifies the encoding of the language provided by the <i>request.attr.language</i> property.
Example	iso639 (allowed values: iso639, iso3166, rfc1766)

Name	request.attr.editLanguage
Description	Extracts the edit language of the user which will be used within TTS PERFORMANCE SUITE. This value overrides the value of <i>request.attr.language</i> .
Example	constant.en-us (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	request.attr.editLanguage.encoding
Description	Extracts the encoding of the edit language provided by property <i>request.attr.editLanguage</i> .
Example	Example: iso639 (allowed values: iso639, iso3166, rfc1766)

Name	request.attr.interfaceLanguage
Description	Extracts the interface language of the user which will be used within TTS PERFORMANCE SUITE. This value overrides the value of <i>request.attr.language</i> .
Example	constant.en (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	request.attr.interfaceLanguage.encoding
Description	Extracts the encoding of the interface language provided by property <i>request.attr.interfaceLanguage</i> .
Example	iso639 (allowed values: iso639, iso3166, rfc1766)

Name	request.attr.groupMembership
Description	<p>Extracts the group membership of a user. The determined group names serve as input values to identify maintenance and process roles within TTS PERFORMANCE SUITE server.</p> <p>The mapping of group names is done in the administration section of the Curator (<i>As sign authorizations to external users</i>). The extracted group names must match the ones set up in the Curator (case sensitive).</p>
Example	parameter.group

Name	request.attr.groupMembership.separator
Description	Defines the way in which the return value of <i>request.attr.groupMembership</i> is separated. If a value list was returned (parameterValues extractor), this property will be ignored.
Example	comma (allowed values: comma, whitespace, semicolon)

Example

```

ttkf.accelerator.user.loginModules =
de.tts.bd.business.login.request.RequestElementsLoginModule
ttkf.accelerator.user.request.auth.value = header.User
ttkf.accelerator.user.request.auth.validator = notEmpty
ttkf.accelerator.user.request.attr.name = header.User
ttkf.accelerator.user.request.attr.uniqueId = header.User
ttkf.accelerator.user.request.attr.firstname = parameter.fname
ttkf.accelerator.user.request.attr.lastname = parameter.lname
ttkf.accelerator.user.request.attr.email = cookie.email
ttkf.accelerator.user.request.attr.language = constant.en
ttkf.accelerator.user.request.attr.language.encoding = iso639
ttkf.accelerator.user.request.attr.groupMembership =
parameterValues.group

```

With this sample configuration, a user is authenticated successfully if the Request header "User" exists and it is not empty. The user profile is built from request parameters, the e-mail address is read from a cookie and the language is set to English. A list of parameter values provides information on the user's group membership.

The following HTTP request authenticates manager and salesman Mario Musterman and populates his profile with information:

```

GET
/webaccess/index.do?fname=Mario&lname=Musterman&group=Sales&group=Ma
nagement
Host: www.ttkfserver.musterfirma.local
User-Agent: Mozilla/5.0
Cookie: email=Mario.Mustermann@musterfirma.local
...

```

Windows-based Single-Sign-On

Purpose of Windows-based SSO is to provide a Single-Sign-On for end users when accessing the TTS PERFORMANCE SUITE WebAccess. The end user is authenticated against an active directory and authorized by permissions mapped to its LDAP-profile.

If authentication fails, an HTTP-401 standard error message is displayed in the browser. In case the user was authenticated successfully, but is missing authorization, an exception is thrown and access is denied.

An advantage of Single-Sign-On, besides automatic login, is the use of existing user information and structures just by mapping LDAP groups to TTS PERFORMANCE SUITE-internal permissions.

This feature uses NTLM v2 and/or Kerberos for negotiation in the authentication process. In the configuration, you can define which protocol is to be used to authenticate the user.

Restrictions

The current implementation restricts the usage of the Windows-based-Single-Sign-On in two ways:

- Current SSO does not provide any fallback mechanism to other login methods as database authentication.
- If authentication fails, the browser automatically sends a HTTP-401 error message combined with a pop-up requesting for basic authentication. The user might login with valid credentials, even though the Single-Sign-On has failed (for example due to different domain location).

Requirements

- Windows OS: This feature supports Windows OS only, since the Account lookup works locally and in Active Directory via Win32 API.
- Browser: Currently, only Internet Explorer is supported.
- The WebAccess and/or Curator site must be in the Intranet Zone.
- Integrated Windows Authentication must be enabled in the browser preferences.
- Domain: The application server must be in the corresponding domain.

Configuration

To enable the Windows-based-SSO, a few configuration steps are necessary. First, the login module *ttn.bd.login.sso.windows.WindowsSSOLoginModule* must be defined in user service.

Next, the LDAP needs to be specified. Without these bits of information, a successful login is not possible.

Last but not least, the Windows-based-SSO contains a few parameters that must be set.

Name	loginModules.WindowsSSOLoginModule.populator
Description	Specifies the populator of the login module, populating user profile and authorities. If not configured, the default populator (LDAP) is used. Values: <i>ttn.bd.login.sso.authorization.WindowsSSODatabase-AuthoritiesPopulator</i>

Name	login.sso.windows.principalFormat
Description	<p>Specifies the name format for the principal.</p> <p>Values:</p> <p><i>fqn</i> - Fully qualified names, such as domain\username. When unavailable, a SID is used. (default)</p> <p><i>sid</i> - SID in S-format.</p> <p><i>both</i> - Both a fully qualified name and a SID in the S- format. The fully qualified name is placed in the list first.</p>

Name	login.sso.windows.roleFormat
Description	<p>Specifies the name format for the role.</p> <p>Values:</p> <p><i>fqn</i> - Fully qualified names, such as domain\username. When unavailable, a SID is used. (default)</p> <p><i>sid</i> - SID in S-format.</p> <p><i>both</i> - Both a fully qualified name and a SID in the S- format. The fully qualified name is placed in the list first.</p>

Name	login.sso.windows.allowGuestLogin
Description	<p>Allows guest login. When true and the system's Guest account is enabled, any invalid login succeeds as Guest.</p> <p>Values:</p> <p>true</p> <p><i>false</i> (default)</p>

Name	login.sso.windows.securityFilterProviders
Description	<p>Defines a list of security filter providers.</p> <p>Value: <i>waffle.servlet.spi.NegotiateSecurityFilterProvider</i></p>

Name	login.sso.windows.protocols
Description	<p>A comma-separated list of security protocols supported by the NegotiateSecurityFilterProvider. May be <i>Negotiate</i> or <i>NTLM</i> (or a combination of both).</p> <p>Values: NTLM,Negotiate</p>

Example

```
ttkf.accelerator.user.loginModules=ttn.bd.login.sso.windows.WindowsS
SOLoginModule
ttkf.accelerator.user.login.sso.windows.principalFormat = both
ttkf.accelerator.user.login.sso.windows.roleFormat = both
ttkf.accelerator.user.login.sso.windows.securityFilterProviders =
waff- le.servlet.spi.NegotiateSecurityFilterProvider
ttkf.accelerator.user.login.sso.windows.protocols = NTLM
ttkf.accelerator.user.ldap.defaultprovider.url =
ldap://server.domain.local:port/
ttkf.accelerator.user.ldap.defaultbase.dn =
OU=User,OU=TTS,DC=teamtraining,DC=local
```

```
ttkf.accelerator.user.ldap.defaultmanager.dn =
CN=admin,OU=Sonstige,OU=User,OU=TTS,DC=tts,DC=local
ttkf.accelerator.user.ldap.defaultmanager.password = password
ttkf.accelerator.user.ldap.defaultdialect =
de.tts.bd.business.login.ldap.ActiveDirectoryDialect
ttkf.accelerator.user.ldap.defaultdialect.adsDomain =
teamtraining.local
ttkf.accelerator.user.ldap.defaultdialect.user.filter =
(&(objectClass=user)(userPrincipalName=${ttc:user}@${ttc:adsDomain})
)
```

6.4.5 SAML Single-Sign-On

With SAML, a new SSO (Single Sign-on) variant is supported for the WebAccess and Curator. To activate SAML, you have to make the following settings in the *application-config.properties* file.

Configuration

General

Name	loginModules.SAMLAAuthenticationLoginModule.populator
Description	Specifies the populator of the login module, which populates user profiles and authorities. If not configured, the default populator (LDAP) is used. Possible value: com.tts.serverfoundation.authentication.saml.SAMLAuthoritiesPopulator

Name	loginModules.SAMLAuthoritiesPopulator.groups
Description	Group names.
Example	enduser,author

KeyStore

Name	security.keystore.path
Description	Path to the key store of the service provider. Value: file:/// <path_to_keystore>

Name	security.keystore.password
Description	Password to the key store of service provider.

Name	security.keystore.serviceprovider.key.alias
Description	Alias name for service provider's key store.

Name	security.keystore.serviceprovider.key.password
Description	Password for the SAML key alias.

Name	saml.security.keystore.identityprovider.certificate.path
Description	Path to the identity provider's certificate, to validate responses from identity provider. Value: <i>file:/// <path_to_cert></i>

Name	saml.security.request.sign
Description	If set to true, the requests to the identity providers are signed with the service provider's certificate. Value: true or false (default: false)

Name	saml.security.request.sign.algorithm
Description	Sets the algorithm used to sign the request. Value: algorithm (default: http://www.w3.org/2000/09/xmlsig#rsa-sha1)

Service Provider Configuration

Name	user.saml.config.serviceprovider.name
Description	Service provider name which is used by tts performance suite. Value: <i>TT KnowledgeForce SSO Service Provider</i>

Name	user.saml.config.serviceprovider.id
Description	Service provider identity which is used by tts performance suite. Value: http://<Accelerator_URL>

Name	user.saml.config.serviceprovider.assertion.consumer.service.url
Description	Service provider identity which is used by tts performance suite Value: http://<Accelerator_URL>/index.do

Identity Provider Configuration

Name	user.saml.config.identityprovider.id
Description	ID from Identity Provider Value: <Identity Provider URL>

Name	user.saml.config.identityprovider.auth.request.service.url
Description	Requested Identity Provider URL Value: https://<Identity Provider URL>

SAML Metadata Mapping

Name	user.saml.user.attr.groupMembership
Description	<p>Extracts the group membership of a user. The determined group names serve as input values to identify process and author roles within TTS PERFORMANCE SUITE server.</p> <p>The mapping of group names is done in the administration section of the Curator (<i>Assign authorizations to external users</i>). The extracted group names must match the ones set up in the Curator (case sensitive).</p> <p>Value: GROUP_ATTR</p>

Name	user.saml.user.attr.groupMembership.separator
Description	Defines the way in which the return value of <i>request.attr.groupMembership</i> is separated. If a value list was returned (parameterValues extractor), this property will be ignored.
Example	comma (allowed values: comma, whitespace, semicolon)

Name	user.saml.user.attr.name
Description	Extracts the (login) name of a user.
Example	Parameter.user (extractors returning a value list as <i>parameterValues</i> are not allowed)

Name	user.saml.user.attr.uniqueId
Description	Extracts a unique identifier (e.g., email address or the username again).
Example	parameter.email

Name	user.saml.user.attr.firstname
Description	Extracts the first name of a user.
Example	<i>parameter.firstname</i>

Name	user.saml.user.attr.lastname
Description	Extracts the last name of a user.
Example	parameter.lastname

Name	user.saml.user.attr.email
Description	Extracts the e-mail address of a user.
Example	parameter.email

Name	user.saml.user.attr.language
Description	Extracts the language of the user which will be used as the edit and the interface language within the TTS PERFORMANCE SUITE server.
Example	constant.en (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	user.saml.user.attr.language.encoding
Description	Specifies the encoding of the language provided by the <i>request.attr.language</i> property.
Example	iso639 (allowed values: iso639, iso3166, rfc1766)

Name	user.saml.user.attr.editLanguage
Description	Extracts the edit language of the user which will be used within TTS PERFORMANCE SUITE. This value overrides the value of <i>request.attr.language</i> .
Example	constant.en-us (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	user.saml.user.attr.editLanguage.encoding
Description	Extracts the encoding of the edit language provided by property <i>request.attr.editLanguage</i> .
Example	iso639 (allowed values: iso639, iso3166, rfc1766)

Name	user.saml.user.attr.interfaceLanguage
Description	Extracts the interface language of the user which will be used within TTS PERFORMANCE SUITE. This value overrides the value of <i>request.attr.language</i> .
Example	constant.en (allowed encoding: ISO 639, ISO 3166, RFC 1766)

Name	user.saml.user.attr.interfaceLanguage.encoding
Description	Extracts the encoding of the interface language provided by property <i>request.attr.interfaceLanguage</i> .
Example	iso639 (allowed values: iso639, iso3166, rfc1766)

Additional SAML configuration

Name	user.saml.compress
Description	SAML request should be sent compressed or uncompressed. Value: true false

Name	userlist.workflowcheck
Description	<p>For better collaboration the mechanism for owners and assignees has been refactored. Instead of showing either all external users or none of them, now the external users are filtered based off their persisted permissions.</p> <p>The requirement for this feature is to import all authors from the external identity provider (e.g., ldap), see chapter <i>user import</i>.</p> <p>Default: true</p> <p>Value: true, filter owners and assignees based off persisted permissions false, filter local users, show external users</p>

6.5 Logging

The tts performance suite server uses log4j2 to log to different media, e.g., console or logfile. These logs are very important to monitor the application at runtime and to identify causes of errors, instabilities, or other malfunctions.

By default, the tts performance suite server logs with the log level INFO to the standard out only. To customize the logging behavior a separate `log4j2.properties` file must be included with one of the following methods.

- As a jvm parameter:

```
-Dlog4j.configurationFile=C:\path\to\log4j2.properties
```

- Via catalina.properties:

```
log4j.configurationFile=C:\\path\\to\\log4j2.properties
```

- Provide the parameter via the context parameter `log4jConfiguration` e.g., in the `context.xml` or `server.xml`

```
<Context>
...
<Parameter name="log4jConfiguration"
value="file:///C:/path/to//log4j.application.properties" />
</Context>
```



The last method is recommended. It allows to use different configuration files for Curator and WebAccess and thus e.g., to log to different logfiles respectively.

The following section shows a few examples on how to configure the tts performance suite logging for specific use cases. Additional information can be found in the official log4j2 documentation (see <https://logging.apache.org/log4j/2.x/manual/configuration.html#Properties>)

Example 1

```
monitorInterval=30
rootLogger.level = INFO
rootLogger.appenderRef.stdout.ref = STDOUT
appender.console.type = Console
appender.console.name = STDOUT
appender.console.layout.type = PatternLayout
appender.console.layout.pattern =
%-5p [${web:servletContextName:-}][%d] %c [%t] - %X - %m%n
```

Example 1 shows a sample logging configuration, that prints all log messages of level INFO and above to the standard out using the given pattern. Note that for runtime configuration changes, the `monitorInterval` must be present in the properties file. The application will check for changes in the logging configuration in the specified interval (in seconds).

Example 2


```
appender.rolling.type = RollingFile
appender.rolling.name = RollingFile
appender.rolling.fileName = c:/temp/curator.log
appender.rolling.filePattern = c:/temp/curator-%d{MM-dd-yy-HH}-
%i.log.gz
appender.rolling.layout.type = PatternLayout
appender.rolling.layout.pattern = %-5p [${web:servletContextName:-
}] [%d] %c [%t]
- %X - %m%n
appender.rolling.policies.type = Policies
appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
appender.rolling.policies.size.size=100MB
appender.rolling.strategy.type = DefaultRolloverStrategy
appender.rolling.strategy.max = 5
logger.upload.name = com.tts.document.upload
logger.upload.level = DEBUG
```


Example 2 shows how to add an additional logger. In this example, it logs all log messages from the package `com.tts.document.upload` with level DEBUG and above to separate files using a rolling file appender.

6.6 Version control and Workflow service

6.6.1 Version control

The Release service provides version control for documents and their content. Any change to a document's properties or its content results in a new version. When combined with the Workflow engine, transitions of the workflow status are version controlled as well.

 You can roll back and delete a version, as well as compare it with other revisions or with the current version.

 With version control enabled, the hard disk space used by the repository and the size of the database will increase significantly. For further details, see the hardware requirements section.

Service name: `releaseService` (**maintenanceInterval:10**)

Name	<code>revisionDeleteEnabled</code>
Description	Allows for the deletion of document revisions. Possible values: true false (default)

Name	<code>maxUpdates</code>
Description	Defines the maximum number of updated documents within the maintenance interval. Possible values: Any positive integer 50 (default) 0 (all buffered)

Name	<code>updateAllOnStartup</code>
Description	Defines whether to recalculate the latest released version for all documents

	<p>on startup or not.</p> <p>Possible values:</p> <p>true</p> <p>false (default)</p>
--	--

6.6.2 Workflow service

The Workflow service allows for creating and administering workflows that are based on document types. In addition to that, workflow transitions are permission controlled through assignable author roles.

Workflow functions

Workflow functions are actions that will be executed when a document's workflow status is changed. To define which functions are triggered by which transition, use the workflow administration section (>*Settings* >*Operational structuring* >*Workflows*). Read on for a brief description of the currently available functions:

Name	IncreaseVersionFunction
Description	Increases the version of the document.

Name	NotifyOwnersFunction
Description	Sends e-mails to the document's owner and assignee when the workflow status changes.

Name	NotifyAssigneeOnlyFunction
Description	Sends an e-mail to the document's assignee when the workflow status changes.

Name	NotifyOwnerOnlyFunction
Description	Sends an e-mail to the document's owner when the workflow status changes.

Name	NotifyMrolesFunction
Description	Sends e-mails to all role members of document's target status after a status change.

Name	CleanupVersionsFunction
Description	All superfluous versions of a document will be deleted upon calling this function, keeping only versions that have been 'published' at some point, as well as all versions that are newer than the last published version.

Name	RevokePublishingFunction
Description	Removes the release flag of a document so the document won't appear in the documentation portal.





For an in-depth documentation of the version control and workflow service features, please refer to the Administrator manual.

6.7 Cache service


This service provides a distributed caching facility. The use of caches plays a key role in achieving high performance by minimizing the number of roundtrips to the database. Distributed caches are used to share data with other applications, for instance a WebAccess.

Changes made to a distributed cache are reflected in each application, even if they are running on different machines.

 **Caching may fail if the server has multiple network adapters attached and at least one adapter is disabled or not working. In this case, replace "localhost" with the network adapter's IP in the configuration you want to use. If the WebAccess and the Curator are running on the same server, 127.0.0.1 should be used**

 You might face some warning messages concerning the buffer size of the operating system:

```
WARN [2012-09-26 16:25:34,825] [Startup] org.jgroups.protocols.UDP
[main] - send buffer of socket java.net.DatagramSocket@1e247e2 was
set to 640KB, but the OS only allocated 131.07KB. This might lead to
performance problems. Please set your max send buffer in the OS
correctly (e.g. net.core.wmem_max on Linux)
```

 In this case, you need to adapt the OS settings, since those problems may result in performance issues. Please ask your system administrator to help you with these settings.

Service name: cache

Name	cacheDomain
Description	<p>Defines the name of the domain in which this application node is located.</p> <p>Possible values:</p> <ul style="list-style-type: none"> workbench docportal

Name	cachePreload
Description	Enables preloading via bootstrap loader. Possible values: true (default) false
Name	peer.discovery
Description	Sets either multicast or unicast cache peer discovery mode. Possible values: multicast (default) unicast

Name	peer.listener.host
Description	Defines the host name on which the listener is listening.
Example	localhost

Name	peer.listener.port
Description	Sets the port on which the listener is listening (necessary in unicast mode only).
Example	7800

Name	peers.multicastGroupAddress (multicast only)
Description	Sets the address of the multicast group this application should join in order to receive cache information.
Example	228.8.8.8

Name	peers.multicastGroupPort (multicast only)
Description	Defines the port of the multicast group this application should join in order to receive cache information.
Example	45566

Name	peers.timeToLive (multicast only)
Description	<p>Specifies how far the packages are propagated (0-255).</p> <p>Possible values:</p> <ul style="list-style-type: none"> 0 - the same host 1 - the same subnet (default) 32 - the same site 64 - the same region 128 - the same continent 255 - unrestricted

Name	peers.workbench (unicast only)
Description	Comma-separated list of host names with ports of peers located in the Curator cache domain.
Example	workbench:7800

Name	peers.docportal (unicast only)
Description	Comma-separated list of host names with ports of peers located in the WebAccess cache domain.
Example	webaccess1:7800,webaccess2:7800



The EHCACHEProvider offers additional configuration options. Please refer to project's website: <http://ehcache.sourceforge.net/documentation/configuration.html> (section ehcache.xml and Other Configuration Files)



If running on an OS with active IPv6, be sure to start the application server with IPv4 preference by means of the JVM parameter "-Djava.net.preferIPv4Stack=true".

6.8 Repository service

The repository service provides access to the document repositories where the content of each document is stored. When uploading document content, the files are first spooled to a local directory (spool store). Within a defined maintenance interval, the files will then be uploaded to their repositories asynchronously.

Service name: repository (maintenanceInterval:5)

Name	repository.maintenance.interval
Description	Verifies the availability of the specified document repositories in an interval given in seconds. Possible value: any positive integer value, e.g., 300

Name	spool.maxRetries
Description	Defines how often a failed upload will be retried before the document is marked as "upload failed". Possible value: any positive integer value, e.g., 5


Name	repository.url
Description	S3-URL pointing to the repository behind MinIO. The URL contains bucket and root path. Possible value: S3-URL, e.g., s3://<bucket>/<path>

Name	repository.endpoint
Description	The local address of the MinIO server. No trailing slashes allowed. Possible value: URL, e.g., http://localhost:9000

Name	repository.endpoint.external
Description	The address of a used load balancer or proxy. No trailing slashes allowed. Possible value: URL, e.g., http://proxy

Name	repository.accessKeyId
Description	The accessKeyId of MinIO.


Name	repository.secretKey
Description	The secretKey of MinIO.

 **By default, there exist repository providers for file system and URL. Providers for other kinds of repositories are available upon request.**

6.9 Notification service

The Notification service automatically sends e-mails when certain events occur in the lifecycles of monitored documents. E.g., users may set up to be notified when documents they are interested in get modified.

The service queues all e-mails and sends them periodically within a defined interval (maintenance interval).

 This service is disabled by default.

Service name: email (maintenanceInterval:30)

Name	enabled
Description	Enables or disables the service. Possible values: true false (default)

Name	mail.smtp.host
Description	Sets the name or IP address of the SMTP server.
Example	smtp.mymailserver.com

Name	mail.smtp.port
Description	Sets the port of the SMTP server.
Example	25

Name	mail.smtp.auth
Description	<p>True if the provided SMTP server requires authentication.</p> <p>Possible values: true (default) false</p>

Name	mail.smtp.user
Description	Defines the username to authenticate against the SMTP server.
Example	tts

Name	mail.smtp.pass
Description	Defines the password to authenticate against the SMTP server.
Example	ttpass

Name	mail.smtp.starttls.enable
Description	<p>Enables STARTTLS command (if supported by the SMTP server) to switch the connection to a TLS-protected connection before issuing any login commands.</p> <p>Note that an appropriate trust store must be configured so that the Producer will trust the server's certificate. Disabled by default.</p> <p>Possible values: true false (default)</p>

Name	mail.smtp.connectiontimeout
Description	Represents the connection timeout in milliseconds. Possible value: any positive integer, e.g., <i>5000</i>

Name	mail.smtp.timeout
Description	Represents the timeout to send data in milliseconds. Possible value: any positive integer, e.g., <i>30000</i>

Name	mail.smtp.sendpartial
Description	Sends the e-mail even if one of the recipients is invalid. Possible values: true (default) false

Name	default.email.dispatcher
Description	To specify the e-mail address of the sender.
Example	any valid e-mail address, e.g., <i>curator@domain.com</i>


Name	default.email.bcc
Description	Defines an e-mail address that is used for blind carbon copy messages.
Example	any valid e-mail address, e.g., <i>bcc@domain.com</i>



For further information about possible parameters, see the following website:
<http://java.sun.com/products/javamail/javadocs/com/sun/mail/smtp/package-summary.html>

6.10 Language service

Language-related functions in the tts Server are provided by the Language service. Languages are expressed either as lowercase ISO 639 language codes (interface language), or as RFC 1766 compliant language tags (edit language).

 **The server's license controls which languages are available. The settings below are used to restrict the list of available languages. You cannot add languages that are not covered in the license.**

Name	available.edit.languages
Description	<p>Restricts the number of available edit languages defined in the license file. Possible values:</p> <p>* = no restriction</p> <p>Comma separated RFC 1766 language tags, e.g., <i>de-de, en-us</i></p>

Name	default.edit.language
Description	Sets the default edit language in RFC 1766 format.
Example	de-de

Name	available.interface.languages
Description	<p>Sets the available interface languages in ISO 639 format.</p> <p>Possible values:</p> <p>Comma separated ISO 639 language codes, e.g., <i>de, en, es</i></p>

Name	default.interface.language
Description	Sets the default interface language in ISO 639 format.
Example	<i>de</i>

6.11 Template service

The WebAccess is basically a smart template engine that can be configured via the template service. All relevant web pages of the WebAccess may be customized to match a customer's requirements, e.g., with regard to the corporate design.

Configurations concerning customer-specific behavior of user login, handling of process, and maintenance roles (standard and preview mode) are also managed by this service.



Templates are defined in a file named *templates.xml* located in `$TTPS_HOME/WEB-INF`. tts provides services to support you in defining a custom portal layout and behavior.

Please contact your Key Account Manager to receive an offer.



The experienced tts customizing team will be happy to offer professional customizing services.

Service name: templates

Name	config
Description	Defines where the service configuration file is located. Possible value: <i>/WEB-INF/templates.xml</i>

Name	configCustom
Description	Sets the path including a wildcard for all customer configurations. Possible value: <i>/WEB-INF/templates.*.xml</i> , e.g., <i>/WEB-INF/templates.customername.xml</i>

Name	configAutoReload
Description	Enables automatic configuration reload when the configuration file has been changed. Possible values: true false (default)

Name	emptyRolesAllowed
Description	<p>Enables login to the WebAccess even if a user has no process roles assigned or there are no roles supplied to the WebAccess.</p> <p>Possible values: true (default) false</p>

Name	defaultRoles
Description	<p>Comma separated list of technical names of process roles to display if a user has no process roles assigned. Works only if the property <i>emptyRolesAllowed</i> is set to false.</p> <p>Possible values: any valid technical names of process roles, e.g. <i>worker</i>, <i>manager</i></p>

Name	emptyRolesAllowedOnPreview
Description	<p>Allows preview for a user to whom no process roles were assigned.</p> <p>Possible values: true (default) false</p>

Name	defaultPreviewRoles
Description	<p>Comma separated list of technical names of process roles to display on preview if a user has no process roles assigned. Works only if the property <i>emptyRolesAllowed</i> is set to <i>false</i>.</p> <p>Possible values: any valid technical names of process roles, e.g. <i>worker</i>, <i>manager</i></p>

Name	ignoreMaintenanceRolesOnPreview
Description	<p>Defines whether the maintenance roles are ignored on preview or not by filtering objects visible to a user.</p> <p>Possible values:</p> <p>true (default)</p> <p>false</p>

Name	userSessionTimeout
Description	<p>Sets a session timeout (in seconds) for accessing the WebAccess.</p> <p>Possible values: any positive integer, e.g., 1800</p>

Name	previewSessionTimeout
Description	<p>Sets a session timeout (in seconds) for accessing the WebAccess preview within Curator.</p> <p>Possible values: any positive integer, e.g., 600</p>

6.12 Scheduler service

This service handles any asynchronous job execution, which includes the maintenance jobs of all other services.

6.13 Configuration service

This service offers settings for the administration of multiple configurations used in Curator and WebAccess.



For more information about multiple configurations, please see the Design Manager guide.

Service name: config

Name	defaultConfiguration
Description	<p>Defines the default configuration when there are multiple configurations.</p> <p>Possible value: name of the configuration, such as <i>initial</i></p>

Name	portalConfigurations
Description	Defines the names of available portal customizings in a comma separated list. These pieces of information are used to provide a list box of customizings within the preview, allowing users to switch the customizing.
Example	initial, tts, customizing1, customizing

Name	mapping.browser.\$BrowserName
Description	<p>Defines the configuration mapped to the given browser name, where \$BrowserName stands for the name of the browser.</p> <p>Sample value:</p> <pre>mapping.browser.OPERA11 = desktop_config mapping.browser.MOBILE_SAFARI = mobile_config</pre> <p>Possible values (snippet):</p> <pre>IE, IE11,IE10,IE9, EDGE,EDGE14 EDGE_MOBILE IEMOBILE10, IEMOBILE11 FIREFOX, FIREFOX48 FIREFOX_MOBILE SAFARI MOBILE_SAFARI CHROME,CHROME51 CHROME_MOBILE</pre>

Name	mapping.operatingSystem.\$OperatingSystemName
Description	<p>Defines the configuration mapped to the given operating system name, where \$OperatingSystemName stands for the name of the OS.</p> <p>Sample value:</p> <pre>mapping.operatingSystem.WINDOWS = desktop_config mapping.operatingSystem.IOS = mobile_config</pre> <p>Possible values (snippet):</p> <pre>WINDOWS, WINDOWS_10, WINDOWS_8, WINDOW_7 ANDROID, ANDROID6, ANDROID6_TABLET, ANDROID5, ANDROID5_TABLET WEBOS, PALM IOS MAC_OS MOBILE_SAFARI</pre>



More information on how to use the browser and operating system-specific configurations can be found in the user manual and the administrator's guide.

6.14 Feature service

This service provides a facility to control and configure the behavior of several components (or features) of the application. Because of the complexity of the configuration, this service uses its own configuration file.

Service name: feature

Name	config
Description	<p>Context-relative name of the configuration file for this service or absolute file path introduced by prefix "file:"</p> <p>Default value:</p> <pre>/WEB-INF/feature-config.xml</pre> <p>Possible value for absolute path:</p> <pre>file:/configfiles/feature-config.xml</pre>

6.15 Miscellaneous parameters

These configuration parameters do not belong to a specific service:

Name	ttkf.integrator.collectionContentHandler.customSupportedTypes
Description	<p>Defines supported mime-types for files within collections (e.g., zip-archives). The mime types have to be added in the format {prefix}.{mimetype}={priority}.</p> <p>prefix: ttkf.integrator.collectionContentHandler.customSupportedTypes mimetype: any valid mime-type priority: any natural number greater than 9</p>
Example	<pre>ttkf.integrator.collectionContentHandler.customSupportedTypes.text/javascript =10 ttkf.integrator.collectionContentHandler.customSupportedTypes.text/richtext =11 ttkf.integrator.collectionContentHandler.customSupportedTypes.text/plain =12</pre>

Name	ttkf.server.guide.repository.url
Description	<p>To store Guides, a Guide-Repository is needed. Therefore a file or S3 Url is needed, where the Guides can be stored:</p> <pre>ttkf.server.guide.repository.url=file:///path with read/write access} ttkf.server.guide.repository.url=s3:///path with read/write access}</pre> <p>It is possible to use the \$TTPS_HOME/repository/guide folder, which we should have been created in <i>Installation Environment</i> chapter:</p> <pre>ttkf.server.guide.repository.url=file:///C:/TTS PERFORMANCE SUITE/repository/guide</pre>



See the *Installation Environment* Chapter in the Installation step-by-step for more details.

Name	ttkf.server.guidebase.paging.size
Description	<p>This parameter is optional.</p> <p>Defines how many guides will initially be shown in every guide category. (my/new/favorite/popular).</p> <p>When the user scrolls in a category and reaches the end of the list of currently visible Guides, the next {ttkf.server.guidebase.paging.size} Guides will be loaded.</p> <p>Default: 20 ttkf.server.guidebase.paging.size={x>=20}</p>
Example	ttkf.server.guidebase.paging.size= 100

Name	ttkf.server.queue.failure.delay
Description	<p>This parameter is optional.</p> <p>Defines the delay between the next repetition of a failed command.</p> <p>Default: 100</p>
Example	ttkf.server.queue.failure.delay= 100

Name	ttkf.server.loginTokenLifetimeMin
Description	<p>This parameter is optional.</p> <p>Defines the lifetime of a loginToken in minutes (for WebAccess and Curator).</p> <p>The default value -1 will lead to an infinite lifetime.</p> <p>Default: -1</p>
Example	ttkf.server.loginTokenLifetimeMin = 10

Name	ttkf.server.otp.expiration.seconds
Description	This parameter is optional. Defines the time in seconds until <i>One Time Passwords (OTPs)</i> expire. Default: 60
Example	ttkf.server.otp.expiration.seconds=30

Name	ttkf.server.content.security.policy.restrictSVG
Description	This parameter is optional. If set to true, a content-security-policy is added as a HTTP response header to mitigate the threat of XSS attacks in SVG files. Default: true
Example	ttkf.server.content.security.policy.restrictSVG=false

Name	ttkf.server.content.security.policy.restrictXML
Description	This parameter is optional. If set to true, a content-security-policy is added as a HTTP response header to mitigate the threat of XSS attacks in XML files. Default: true
Example	ttkf.server.content.security.policy.restrictXML=false

6.16 Search service

6.16.1 Solr Search Service

The search service is used to provide multilingual search capabilities for the most important objects in the tts server. You can search in the titles and properties of the following objects:

- processes
- topics
- courses
- documents
- glossary entries

In the WebAccess you can find additionally:

- guides

In the Curator you find additionally:

- users

The search service is based on Apache Solr, a text search engine written entirely in Java. In order to find results quickly and accurately, the metadata of the mentioned objects are stored in separate indexes. These are implementation details from Solr.

The indexes are created automatically after database initialization or after tts server updates (containing internal database updates).



A full recreation and re-indexing can be forced over the installation page. There exists a link "Rebuild search index", which will rebuild the entire index asynchronously. Administrator privileges are necessary.

Solr mandatory parameter

Name	<i>ttkf.server.search.server.url</i>
Description	This mandatory parameter defines how the Curator and WebAccess can reach the Solr server. <i>ttkf.server.search.server.url={full context path to the Solr instance}</i>
Example	<i>ttkf.server.search.server.url=http://127.0.0.1:8983/solr/tts-server</i>

6.16.2 Search parameter

Name	<i>ttkf.server.search.max.result.size</i>
Description	This optional parameter defines the maximum number of unfiltered search results which should be returned by a query. By default, each search request will be limited to 10.000 results. Default: 10.000 <i>ttkf.server.search.max.result.size={Positive integer}</i>
Example	<i>ttkf.server.search.max.result.size=5000</i>

Name	<i>ttkf.server.search.disjunction.tie</i>
Description	<p>This optional parameter specifies the <i>tie breaker</i> for the <i>DisMax</i> parser in the SolR search engine.</p> <p>When searching multiple fields, this factor defines the impact that additional matches will have on the score of a result item. For example, when a query matches the title and the description of a document, with the best match found in the title, the score of the additional match in the description will be multiplied by the tie factor and added to the score of the best match in the title of the document.</p> <p>With a tie factor of 0.0, additional matches will be ignored.</p> <p>With a tie factor of 1.0, the overall score of a result item will be the sum of all field matches. Usually, the tie factor should be much less than 1.0.</p> <p>Default: 0.5</p> <p><i>ttkf.server.search.disjunction.tie</i>={Between 0 and 1}</p>
Example	<i>ttkf.server.search.disjunction.tie</i> =0.0



See https://solr.apache.org/guide/8_9/the-dismax-query-parser.html for more details.

6.16.3 Highlighting parameter

The following configuration parameters for the highlighting of query results are optional. Each highlighting parameter has a direct equivalent in the SolR syntax.



See https://solr.apache.org/guide/8_9/highlighting.html for more details.

Name	<i>ttkf.server.search.highlighting.snippets</i>
Description	<p>Maximum number of highlighted snippets per field.</p> <p>Default:1 <i>ttkf.server.search.highlighting.snippets</i>={Positive integer}</p>
Example	<i>ttkf.server.search.highlighting.snippets</i> =2

Name	<i>ttkf.server.search.highlighting.fragsize</i>
Description	Size in characters for highlighted fragments. Default: 256 <i>ttkf.server.search.highlighting.fragsize={Positive integer}</i>
Example	<i>ttkf.server.search.highlighting.fragsize=256</i>

Name	<i>ttkf.server.search.highlighting.simple.pre</i>
Description	HTML markup to insert at the start of a match in the fragment. Default: <code></code> <i>ttkf.server.search.highlighting.simple.pre={HTML markup}</i>
Example	<i>ttkf.server.search.highlighting.simple.pre=<p></i>

Name	<i>ttkf.server.search.highlighting.simple.post</i>
Description	HTML markup to insert at the end of a match in a fragment. Default: <code></code> <i>ttkf.server.search.highlighting.simple.post={HTML markup}</i>
Example	<i>ttkf.server.search.highlighting.simple.post=</p></i>

Name	<i>ttkf.server.search.chunksize</i>
Description	Defines the number of elements, which shall be indexed by Solr at the same time. It is an optional parameter. Default: 100
Example	<i>ttkf.server.search.chunksize = 50</i>

Name	<i>ttkf.server.search.epss.max.group.count</i>
Description	Defines the maximum of groups Solr should return in a search response. This is an optional parameter. Default: 50
Example	<i>ttkf.server.search.epss.max.group.count = 100</i>

Name	<i>ttkf.server.search.epss.group.limit</i>
-------------	---

Description	Defines the maximum limit of rows Solr should return in a search response. This is an optional parameter. Default: 1000
Example	<code>ttkf.server.search.epss.group.limit = 100</code>

Name	<code>ttkf.server.search.optimize.solr.index.interval</code>
Description	Defines the interval to optimize the solr search index. Value is given in hours, default value is 24 hours.
Example	<code>ttkf.server.search.optimize.solr.index.interval = 24</code>

6.17 Security configuration

Password security

Name	<code>ttkf.server.security.password.reset.active</code>
Description	<i>Default: false</i> <i>Possibility: true false</i> This is the global option to enable (true) or disable (false) the resetting of passwords, both initial passwords or expired ones.

Name	<code>ttkf.server.security.password.history.size</code>
Description	<i>Default: 4</i> <i>Possible value: 1-N</i> Specifies the number of passwords in the history. A new user password must be different from the ones contained in the history. Each user has their own history.

The server provides restrictions for new passwords. The following restrictions exist:

- Minimum character length (default 7)
- Username may not be contained in password
- First and last name may not be contained in password.
- RegEx condition. Default settings are:
 - At least one lower case character.
 - At least one upper case character.
 - At least one number.

- At least one special character.

By default, 3 of the 4 conditions must be met.

- The last N passwords of a user cannot be re-used as new password.

The following optional settings exist to configure the password restrictions:

Name	ttkf.server.security.password.min.length
Description	Default: 7 Minimum character length for a new password.

Name	ttkf.server.security.password.regex.condition.enable
Description	Default: true Possibility: true false

Activate the regex conditions for new passwords. The conditions are:

- At least one lower case character.
- At least one upper case character.
- At least one number.
- At least one special character.

Name	ttkf.server.security.password.regex.condition.min.matches.to.pass
Description	Default: 3 Possible value: 0-4 At least n regex condition(s) must match for a new password. (see the previous setting for more information about regex conditions)

Name	ttkf.server.security.password.expiration.days
Description	Default: 90 Possibility: -1 to 365 Specifies for how long (days) a password is valid. -1 disables the expiration check.

Name	ttkf.server.disable.clickjacking.protection.headers
Description	<p>Default: all</p> <p>Possibility: all none x-frame-options frame-ancestors</p> <p>To prevent clickjacking inside of our webapps, you can prevent that our webapps (WebAccess/Curator) are embedded in other webpages.</p> <p>If it is set to none, both the X-Frame-Options and Content-Security-Policy: frame-ancestors headers will be set (see <code>ttkf.server.content.security.policy.frame.ancestors</code> for more information about these headers).</p> <p>If it is set to x-frame-options, only the Content-Security-Policy: frame-ancestors header will be set.</p> <p>If it is set to frame-ancestors, only the X-Frame-Options header will be set.</p> <p>If it is set to all, no header will be set. This means that the clickjacking protection will be disabled completely.</p> <p>See <code>ttkf.server.content.security.policy.frame.ancestors</code> on how to allow embedding in whitelisted pages while clickjacking protection is enabled.</p>

Name	ttkf.server.content.security.policy.frame.ancestors
Description	<p>Default:</p> <p>Possibility: comma separated list of domains</p> <p>The Internet Explorer supports only an old header, which is restricted to one domain. That means, if a list of domains is maintained, the Internet Explorer allows only the first domain. All other browsers allow the whole list of domains.</p> <p>If clickjacking protection is enabled but <code>ttkf.server.content.security.policy.frame.ancestors</code> is not set, our webapps send following headers:</p> <p>X-Frame-Options: SAMEORIGIN</p>

	<p>Content-Security-Policy: frame-ancestors 'self'</p> <p>If clickjacking protection is enabled and <code>ttkf.server.content.security.policy.frame.ancestors</code> is defined:</p> <p>X-Frame-Options: ALLOW-FROM <first entry from <code>ttkf.server.content.security.policy.frame.ancestors</code>></p> <p><i>Content-Security-Policy: frame-ancestors 'self'</i> <first entry from <code>ttkf.server.content.security.policy.frame.ancestors</code>> <second entry from <code>ttkf.server.content.security.policy.frame.ancestors</code>> <third entry></p> <p>X-Frame-Options will be used as fallback for Internet Explorer. All other Browser ignore X-Frame-Options when Content-Security-Policy: frame ancestors is set.</p>
--	--

Miscellaneous

Name	ttkf.server.security.download.attachment.mimetypes
Description	<p>Default: no default (parameter is optional) Possibility: comma separated list of mimetypes without any blanks</p> <p>The most common Microsoft Office mimetypes will be delivered with the Content-Disposition header set to "attachment" for any download requests from IE11. If any more mimetypes should be delivered this way they can be added by using this parameter.</p>

6.18 Dashboard

The Dashboard provides a graphical overview of the usage data in the portal, which is collected and staged by Piwik. It is available via the "Dashboard" link in the portal (provided it is contained in the license and the user has sufficient rights).

6.18.1 License and user rights

The dashboard is a license component and thus must be contained in the license to be available.

Further the user must be assigned an author role which has the right to see the dashboard. This can be configured in the edit author role dialogue under the tab "portal".

If the portal is available without login, the dashboard and the corresponding link are not visible by default.

6.18.2 Configuration of the Piwik connection

The connection to the Piwik server can be configured in the WEB-INF/templates.CUSTOM.xml used in the scenario.

The following values can be set:

```
<property name="piwikServer">piwik-server-url</property>
<property name="piwikToken">789ASDASD</property>
<property name="piwikSiteIdr">17</property>
<property name="piwikCustomDimensionContentType">1</property>
<property name="piwikDebug">>false</property>
```

6.18.3 Specific error pages

If the dashboard is not available due to missing authorization or the like, an error page is displayed instead. This page is located at templates/dashboard/dashboard.error.jsp and distinguishes the following cases:

- The dashboard is not contained in the license.
- The user does not have sufficient rights to see the dashboard.
- The dashboard was called although the portal is available without login (authorization failure).

6.19 User Import Process

User import files (XSD file)

You can find this XSD file available at a public url in your deployed Curator. See resources.

User import files (XML file)

The format has changed and you can verify the validation of the configuration file with the xsd. The difference with the previous format is that only already existing process roles and maintenance roles will be added to an user. Not existing roles will be silently ignored.

User import files (Excel file)

We now manage the domain field differently. There are 3 values authorized:- local, when the user is authenticated with tts performance suite

- ldap, when the user is authenticated with ldap
- saml, when the user is authenticated with saml

Known issues and limitation

No warnings in reports.

Currently no hints are available in the generated report, if a user is successful created but with warnings.

For example:

- Role assignment failed, because the role does not exist

Excelfile

- Username should not contain any whitespaces. Also, no leading or trailing whitespaces.
- Excelfile should not contain the same username more than once. Username which are only distinct in upper or lower case will be handled as same username.

When values in field are not supported, you get description of errors in the excel report.

Resources

[http\(s\)://<ip>:<port>/curator/schemas/user/excelimport/v1/schema.xsd](http(s)://<ip>:<port>/curator/schemas/user/excelimport/v1/schema.xsd)

6.20 Overview about the login modules

Login module	Populators (default)	Configuration parameter	Login behavior	Authentication
DefaultLoginModule	-	-	user from database	username/password
SAMLAAuthenticationLoginModule	SAMLDefaultAuthoritiesPopulator	com.tts.serverfoundation.authentication.saml.SAMLAAuthenticationLoginModule	user from database/temporary runtime user	SAML
	SAMLRuntimeAuthoritiesPopulator	com.tts.serverfoundation.authentication.saml.populators.SAMLRuntimeAuthoritiesPopulator	temporary runtime user	SAML
	SAMLDatabaseAuthoritiesPopulator (deprecated)	com.tts.serverfoundation.authentication.saml.populators.SAMLDatabaseAuthoritiesPopulator	user from database	SAML
LdapLoginModule	LdapAuthoritiesPopulator	com.tts.serverfoundation.authentication.ldap.LdapLoginModule	user from database/temporary runtime user	LDAP
ExtendedLdapLoginModule (deprecated)	ExtendedLdapAuthoritiesPopulator	de.tts.bd.business.login.ldap.ExtendedLdapLoginModule	temporary runtime user	LDAP
SimpleLdapLoginModule (deprecated)	-	de.tts.bd.business.login.ldap.SimpleLdapLoginModule	user from database	LDAP
WindowsSSOLoginModule	WindowsSSOAuthoritiesPopulator	ttn.bd.login.sso.windows.WindowsSSOLoginModule	user from database/temporary runtime user	Windows SSO
	ExtendedLdapAuthoritiesPopulator (deprecated)	de.tts.bd.business.login.ldap.ExtendedLdapAuthoritiesPopulator	temporary runtime user	Windows SSO

6.21 Adapt message for incompatible server & Producer

In a staged rollout of the tts performance suite Producer during an update there might be instances where for a certain period of time incompatible Producers are still installed on the authors' PCs. If an author now tries to login to the Curator a message is being shown which informs the person of the incompatibility.

As of tts performance suite 2018 this message can be supplemented with custom information. For instance, a custom message can now be shown with a point of contact which helps the author solve this issue, thereby reducing unnecessary helpdesk inquiries.

The message can be adapted on server level in the datastore in the file

```
config/client/version-mismatch-custom.json
```

If the file does not exist yet, create a new one and add e.g., the following to this json-file:

```
{ "message": "Unfortunately, your Producer is not compatible to the
tts performance suite server. Please get in contact with
IT@yourcompany.com to request an update of your Producer." }
```

Currently, only one language is supported.

In case personal information are shown in the message (e.g.: the name of the assigned IT support person) please make sure that you get the approval of those affected. Additionally, please bear in mind that the display of this message does not require a tts performance suite login. Therefore, any sensible information must not be included.

6.22 Configure WebSocket Support for Apache Webserver with AJP

6.22.1 Missing WebSocket Support In AJP

As for now, AJP 1.3 does not support WebSockets. This means, communication via WebSocket is not possible when using AJP between Apache Webserver (e.g., as proxy or reverse proxy) and Apache Tomcat Application Server. This leads to malfunctions within the ribbon (buttons won't be update concerning the current document state) and the user import feature.



The following error in the log files states missing WebSocket support exception is `java.lang.UnsupportedOperationException: HTTP upgrade is not supported by this protocol`

6.22.2 How to configure Apache Webserver?

The following modules must be activated within `httpd.conf`:

```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

Next the urls, which are to be upgraded to websocket protocol must be redirected via rewriting:

```
RewriteEngine On
RewriteCond %{HTTP:Upgrade} =websocket [NC]

RewriteRule /curator/ui/document/(.*)
ws://localhost:8080//curator/ui/document/$1 [P,L]
RewriteRule /curator/user/import/websocket/(.*)
ws://localhost:8080//curator/user/import/websocket/$1 [P,L]
```

6.23 Security Recommendations

This section describes the latest recommendations for installing a tts performance suite Server securely. Any potential security issues that can be solved with infrastructure changes will be listed here.

These recommendations may depend on the specific environment and not be relevant to all customer setups.

6.23.1 HTTP Strict Transfer Security

If the Curator is, for instance, accessible under the SSL-secured URL `https://example.com/curator`, an attacker with access to the network can provide a user with the link `http://example.com/curator` and (given that SSL is no longer used) must not verify their identity and can pretend to be the Curator (e.g. for phishing).

To avoid that, we can send the HTTP header Strict-Transport-Security (HSTS, [read more here](#)) which tells the user's browser that they should only access the Curator (or WebAccess) over https. This can be done with Tomcat by enabling the following filter in the `web.xml`:

```
<filter>
<filter-name>httpHeaderSecurity</filter-name>
<filter-class>
org.apache.catalina.filters.HttpHeaderSecurityFilter
</filter-class>
<async-supported>true</async-supported>
<init-param>
<param-name>hstsMaxAgeSeconds</param-name>
<param-value>31536000</param-value>
</init-param>
<init-param>
<param-name>hstsIncludeSubDomains</param-name>
<param-value>true</param-value>
</init-param>
<init-param>
<param-name>antiClickJackingEnabled</param-name>
<param-value>true</param-value>
</init-param>
<init-param>
<param-name>antiClickJackingOption</param-name>
<param-value>SAMEORIGIN</param-value>
</init-param>
</filter>
```

And un-commenting this *filter-mapping*:

```
<filter-mapping>
<filter-name>httpHeaderSecurity</filter-name>
<url-pattern>/*</url-pattern>
<dispatcher>REQUEST</dispatcher>
</filter-mapping>
```



max-age

The above max age (31,536,000 s) is equivalent to a year. The following recommendations are given:

- Wikipedia: "sites should set a period of several days or months depending on user activity and behavior".
- HSTS Preload list requires at least 1 year.
- Qualys (SSL Labs): "It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120 days) and ideally to 31536000 (one year)."

Returning a value of 0 will cause the Browser to forget the HSTS entry.



hstsIncludeSubDomains

If there are other web services on the same domain which do not use SSL, you may wish to set `hstsIncludeSubDomains` to `false`.

One possibility is to ramp up the `hstsMaxAgeSeconds` value, checking if any problems occur.



antiClickJackingEnabled

This option is not directly related to HSTS but prevents foreign scripts from being displayed or executed within the Curator or WebAccess, notably in iframes. Please disable this by removing the `antiClickJackingOption` and setting `antiClickJackingEnabled` to `false` in the following situations:

- If using an installation older than 2020 r2
- If using the QuickAccess Web

In addition to `Strict-Transport-Security`, this filter sets these important headers:

- `X-Content-Type-Options: nosniff`
Prevents the browser from overriding the MIME-type and potentially executing .exe disguised as images.
- `X-Frame-Options: SAMEORIGIN`
Prevents the site being embedded in an iframe on an external site (and prevent click-jacking).
- `X-XSS-Protection: 1; mode=block`
Prevents cross-site-scripting in some browsers.



These headers may alternatively be set in a reverse proxy but the configuration is out-of-scope of this document; please consult the relevant documentation for setting headers in your reverse proxy.

6.23.2 HTTP Request Smuggling

- This only applies to **Tomcat** installations behind a **reverse proxy** using HTTP/1.1.

On HTTP/1.1, it is possible for web servers to send and receive packages using a length or using chunks of data. They determine this by using headers:

- Content-Length: X
- Transfer-Encoding: chunked

If, however, you send both headers, some web servers will behave differently from others. Technically the Transfer-Encoding header should take priority. However, some web servers don't support this, and others can be tricked into not using it.

If you are using two servers in a chain (e.g., in a reverse proxy scenario), it can be possible to 'smuggle' a second request by manipulating the headers and appending our own request. The smuggled request is then interpreted as a second request from the original user, e.g.

```
POST /login HTTP/1.1
Host: tt-s.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 62
Transfer-Encoding: chunked

16
login=xxx&password=xxx
0
```

```
GET /404 HTTP/1.1
X-Foo: bar
```

The mitigation will depend on the setup and web servers involved. Two possibilities are:

6.23.2.1 Disable Keep-Alive

Separate requests from the same Producer will create new connections. This will increase over-head. To disable Keep-Alive, add `maxKeepAliveRequests="1"` to the `<Connector />` tag.

6.23.2.2 Use HTTP/2

HTTP/2 no longer uses Content-Length preventing ambiguity between web servers.

Even if it not possible to use HTTP/2 between the Reverse Proxy and the users, HTTP/2 can be used in the back end between the Reverse Proxy and Tomcat. To do this in Tomcat, please add the following to your `<Connector>` tag:

```
<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol"/>
```

e.g.

```
<Connector  
connectionTimeout="20000"  
port="8080"  
protocol="HTTP/1.1"  
redirectPort="8443"  
useBodyEncodingForURI="true">  
<UpgradeProtocol classNa-  
me="org.apache.coyote.http2.Http2Protocol"/>  
</Connector>
```



You will have to refer to the documentation for your reverse proxy for using HTTP/2.