

tts performance suite

Creator 2.0.x Installation
Manual - tts performance
suite 2022r2



.....



Contents

1	General Information	3
1.1	Introduction	3
1.1.1	Objective	3
1.1.2	Target audience	3
1.1.3	Prerequisites	3
1.1.4	Contact	3
1.1.5	Structure	3
1.1.6	Conventions	4
1.1.7	Your feedback is highly welcome.....	4
1.2	Application description	4
1.3	System overview and deployment scenarios	4
2	System requirements	5
3	Installation step-by-step	5
3.1	Checklist.....	5
3.2	Installation environment.....	5
3.2.1	MinIO installation.....	5
3.2.2	Connecting the tts performance suite and the Creator	5
3.2.3	Configuring the Creator.....	5
3.2.4	Configuring the steplist template	6
3.2.5	How to make creator.application.properties accessible?.....	7
3.3	Application server settings.....	7
3.3.1	JVM settings.....	7
3.4	Deployment	7
3.5	Post installation steps	9
3.5.1	Test the creator	9
4	Appendix	10
4.1	creator.application.properties reference	10
4.2	Template options reference	11
4.2.1	Example	11
4.2.2	Description	12

1 General Information

1.1 Introduction

1.1.1 Objective

This document describes the requirements of the tts performance suite component named **Creator**, and its installation process.

The goal of this document is to provide an understanding of the server configuration and to guarantee its successful installation. For installation instructions of other tts performance suite components, Curator and WebAccess, refer to the corresponding manuals.

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 application server Apache Tomcat
- Deployment of web applications (WAR, external web application)

 **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 Creator 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 Installation chapter guides you step-by-step through the deployment of the Creator, including defining server configuration. Post installation steps deals with initial administration tasks.


Due to the fact that the Creator is a complex application, many configuration options are provided. A variety of 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 Variables

 Variables are marked with a leading \$

Variable	Description
\$TTPS_HOME	installation directory of tts performance suite
\$TOMCAT_HOME	installation directory of Apache Tomcat

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 Creator consists of a backend component, written in Java (Spring Boot) and a frontend created with Typescript. The backend is responsible for the connection to the curator and the repository (S3/MinIO), the frontend presents the Creator editor to the user.

1.3 System overview and deployment scenarios

The Creator is a standalone application, which is deployed as a separate war file to an application server. Nevertheless, you'll need the tts performance suite installed and accessible to use it.

There are two different deployment scenarios.

The first scenario, which is the minimum way in practice, involves one application server hosting the tts performance suite (Curator and WebAccess) and also the Creator.

The second, recommended, scenario would be to deploy the Creator to its own application server. It differs from the first scenario by being more scalable and dynamic, thanks to using separate application servers plus a load balancer that distributes all requests among the reverse proxies.

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.

2 System requirements



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

3 Installation step-by-step

3.1 Checklist

Before you start deploying the Creator, please ensure that the following requirements are met:

- The installation files are available
- The application server is installed properly
- The MinIO Server is installed and configured
- The tts performance suite is installed

3.2 Installation environment

3.2.1 MinIO installation

For full MinIO installation instructions refer to the tts performance suite installation manual. It is required that the Creator use the same MinIO instance as the Curator.

Create a bucket named "bg-creator", which is the default name. To use another bucket name use the corresponding value within the `creator.application.properties`.

The properties are explained in detail in the appendix.

3.2.2 Connecting the tts performance suite and the Creator

This parameter needs to be added to the `application-config.properties` of the tts performance suite to connect to the Creator:

```
1 ttkf.server.businessguidancecreator.endpoint=https://<creator-domain>:<port>/  
↪ creator/
```

3.2.3 Configuring the Creator

The actual configuration is carried out via a file named `creator.application.properties`. Place it directly in `$TTPS_HOME`.

The following example shows a snippet of a `creator.application.properties` file for the minimum deployment scenario with the creator placed on the same application server as the tts performance suite.

For all available parameters check the `creator.application.properties` reference in the appendix.

```

1 # Internal URL of the Curator
2 creator.curator.internal-address=https://<curator-domain>:<port>/curator
3
4 # External URL of the Curator
5 creator.curator.external-address=https://<curator-domain>:<port>/curator
6
7 # Version of the Curator where creator.curator.internal-address points to
8 creator.curator.version=<curator-version>
9
10 # Seconds until the jwt expires.
11 creator.jwt-max-age = 300
12
13 # Send a keep-alive request every keep-alive-interval seconds.
14 creator.keep-alive-interval = 150
15
16 # Time in seconds to wait before retrying a failed keep-alive
17 creator.keep-alive-retry-delay = 10
18
19 # Internal address of minio server.
20 # Must be exactly the same value as the
21 #   ttkf.server.repository.endpoint.internal
22 # parameter in the Curator configuration.
23 # E.g.
24 # if
25 #   ttkf.server.repository.endpoint.internal=http://127.0.0.1:9000
26 # is used in the Curator, use
27 #   creator.s3.minio.internal-address= http://127.0.0.1:9000
28 # if
29 #   ttkf.server.repository.endpoint.internal=http://localhost:9000
30 # is used in the Curator, use
31 #   creator.s3.minio.internal-address= http://localhost:9000
32 creator.s3.minio.internal-address=https://<minio-domain>:9000
33
34 # External address of minio server
35 creator.s3.minio.external-address=https://<minio-domain>:9000
36
37 # MionIO authentication
38 creator.s3.access-key={MinIO credentials}
39 creator.s3.secret-key={MinIO credentials}
40
41 # Path to the templates directory
42 creator.templates.location=file:/<path-to-templates-directory>

```

3.2.4 Configuring the steplist template

The steplist template is configured using the file referenced by *creator.templates.options* in *creator.application.properties*. The following shows an example configuration.

The settings are described in the *Appendix*.

```

1 steplist:
2   use-informal-auto-texts: true|false
3   recorder:
4     port: <recorderPort>

```

3.2.5 How to make creator.application.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 creator.application.properties directly in the corresponding Creator context file.

To do so, add the *spring.config.location* context parameter in the context file during deployment (see the deployment chapter):

```
1 ...
2 <Parameter name="spring.config.location" override="false" value="file:/C:/tts/
↳ creator.application.properties" />
3 ...
```

3.3 Application server settings

3.3.1 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)

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

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

3.4 Deployment

Usually, tts provides pre-configured .WAR files for the server components.

Starting with Creator, release 2022r2, there's a templates.zip which has to be deployed additionally to the server.

1. Stop Tomcat if it is running.
2. Unzip the templates.zip and copy its content to \$TOMCAT_HOME/
3. Copy *creator.war* to \$TOMCAT_HOME/webapps.
4. Create the *creator.xml* file in the \$TOMCAT_HOME/conf/Catalina/localhost directory.
5. Configure the location of the *creator.application.properties* within this context file.

Example:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3   <Context reloadable="false">
4
5     <Parameter name="spring.config.location" override="false" value="file:/C:/
↪ tts/creator.application.properties" />
6
7   </Context>
```

6. Restart Tomcat.

3.5 Post installation steps

3.5.1 Test the creator

Log in to the Curator

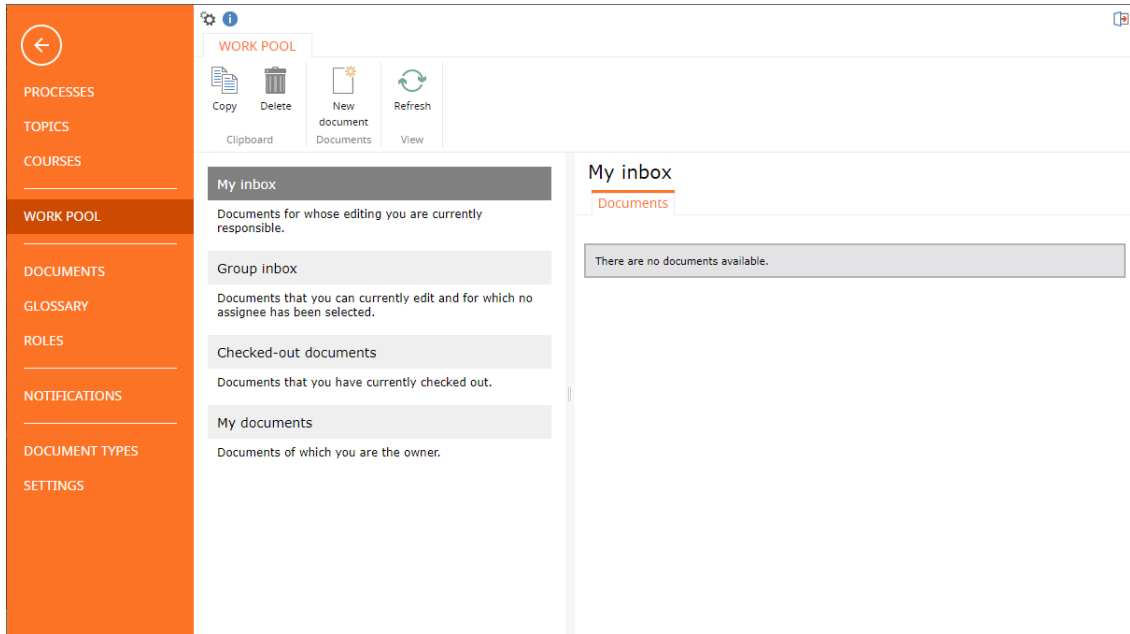


Figure 1: The Curator while being logged-in

When creating a new document, on step three of the wizard, there should be the option "Use the Creator"

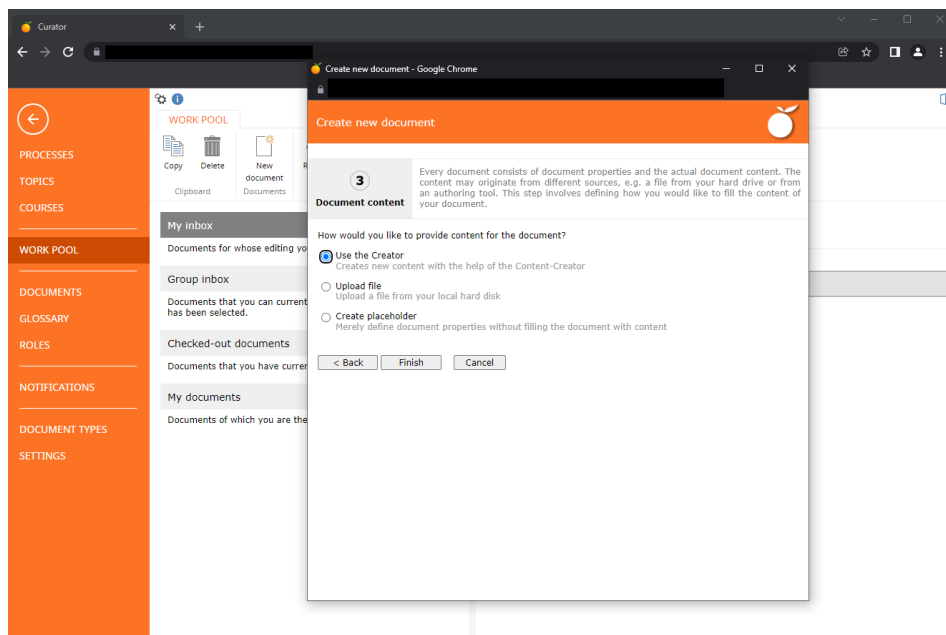


Figure 2: Creating a Creator document

After clicking “Finish” the Creator editor should open with a new document in a separate browser window.

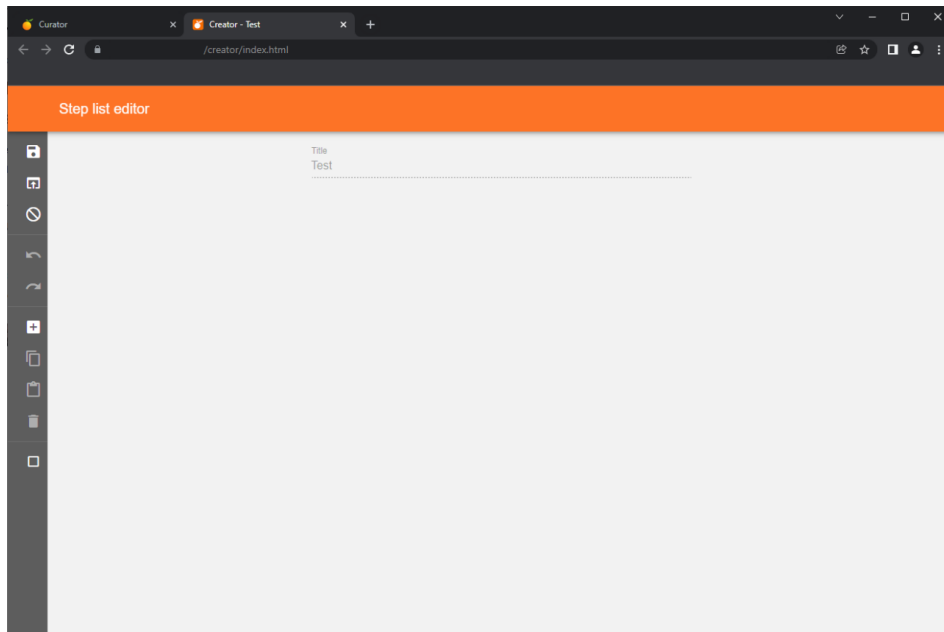


Figure 3: The Creator

4 Appendix

4.1 creator.application.properties reference

Name	Description	Default
creator.curator.external-address	External URL of the Curator.	-
creator.curator.internal-address	Internal URL of the Curator.	-
creator.curator.version	Version of the used Curator. For Example “22.1”.	-
creator.jwt-max-age	Seconds until the jwt expires. Must be $0 < \text{jwt-max-age} \leq \text{Curator JSESSIONID Max-Age}$.	0
creator.keep-alive-interval	Send a keep-alive request every keep-alive-interval seconds. Must be $0 < \text{keep-alive-interval} < \text{Curator JSESSIONID Max-Age}$.	0
creator.keep-alive-retry-delay	Time in seconds to wait before retrying a failed keep-alive request. Must be $0 < \text{keep-alive-retry-delay} < \text{Curator JSESSIONID Max-Age}$.	0
creator.s3.access-key	Access key ID to access s3/minio bucket. If creator.s3.accessKey or creator.s3.secretKey is omitted DefaultCredentialProvider is used to find credentials.	-

Name	Description	Default
creator.s3.arn-kms-key	If an encrypted bucket is used this needs to contain the ARN of the KMS key.	
creator.s3.arn-role	To grant users access to the s3 bucket, temporary credentials are being created using STS AssumeRole. This property must contain the ARN of the parent role that limits the permissions of the temporary credentials. The temporary credentials cannot assume broader permissions than the parent role. The parent role needs to allow s3:ListBucket access on the bucket, and s3:GetObject and s3:PutObject on the prefixed key. If an encrypted bucket is used, it needs to allow kms:Encrypt, kms:Decrypt and kms:GenerateDataKey with the KMS key ARN. arn-role can be omitted if minio is used.	
creator.s3.bucket-name	-	bg-creator
creator.s3.key-prefix	Path used by the creator inside the bucket. Must be empty or end with a '/'.	
creator.s3.minio	Minio options. Must be omitted if AWS S3 buckets are used.	-
creator.s3.minio.external-address	External address of minio server.	-
creator.s3.minio.internal-address	Internal address of minio server.	-
creator.s3.region	-	us-east-1
creator.s3.secret-key	Secret access key to access s3/minio bucket. If creator.s3.accessKey or creator.s3.secretKey is omitted DefaultCredentialProvider is used to find credentials.	-
creator.templates.external-upload	Enable the external document Upload in the template selection if mode is 'create'. Only 'true' is true.	true
creator.templates.location	Path to the given templates. e.g. file:/	-
creator.templates.options	Path to the one and only templates options yaml file. e.g. file:/	-
creator.web-log-level	Only log entries with a level equal or 'above' will be logged on the server. Must be debug, info, warn or error.	info
logging.config	Path to custom log4j2.xml configuration file. If not set a default log4j2.xml is used.	-
server.servlet.context-path	-	/creator

4.2 Template options reference

4.2.1 Example

```

1 steplist
2   use-informal-auto-texts: true|false

```

```
3 recorder
4   port: <recorderPort>
```

4.2.2 Description

Name	Description	Default
steplist.use-informal-auto-texts	Set to true if you want to generate informal auto text when recording	false
steplist.recorder.port	Port to connect to the QuickAccess Recorder	10646