

Installation speech analysis software of EML Windows version 2.3



Installation manual for system providers

9/8/2021

Product line neo, version 6.x

The described functions can be used with the following ASC products:

EVOIPneo

EVOLUTIONneo / XXL / eco

EVOflex (country-specific)

Please note that you can always find the most up-to-date technical documentation and product updates in the partner area on our website at <http://www.asctechnologies.com>.

Copyright © 2021 ASC Technologies AG. All rights reserved.

Windows is a registered trademark of Microsoft Corporation. VMware® is a registered trademark of VMware, Inc. All other marks and names mentioned herein may be trademarks of their respective companies.

Contents

1	General information	4
2	Introduction	5
3	Installation	6
3.1	EML Transcription Server Version Windows 2.3.pdf.....	7

General information

In the context of this document ASC represents ASC Technologies AG, its subsidiaries, branch offices, and distributors. An up-to-date overview of the aforementioned entities can be found at <https://www.asctechnologies.com>

ASC assumes no guarantee for the actuality, correctness, integrity or quality of the information provided in the manuals.

ASC regularly checks the content of the released manuals for consistency with the described hardware and software. Nevertheless, deviations cannot be excluded. Necessary revisions are included in subsequent editions.

Some aspects of the ASC technology are described in general terms to protect the ownership and the confidential information or trade secrets of ASC.

The software programs and the manuals of ASC are protected by copyright law. All rights on the manuals are reserved including the rights of reproduction and multiplication of any kind, be it photo mechanical, typographical or on digital data media. This also applies to translations. Copying the manuals, completely or in parts, is only allowed with written authorization of ASC.

Representative, if not defined otherwise, is the technical status at the time of the delivery of the software, the devices and the manuals of ASC. Technical changes without specified announcements are reserved. Previous manuals lose their validity.

The general conditions of sales and delivery of ASC in their latest version apply.

2 Introduction

This manual describes the installation of the audio analysis software EML Transcription Server of the company EML for Windows operating systems to be used with the *neo* recording system. EML Transcription Server allows transcribing audio into text or detecting keywords which can then be searched for.

Audio analysis jobs are configured and administrated in the Audio Analysis module of the application INSPIRATION*neo*.

Transcription

Transcription is based on the LVCSR technology (large vocabulary continuous speech recognition).

The transcription converts audio recordings into text which is then available for analysis. To be able to recognize all words, dictionaries are uploaded to look up the audio data in. As each separate word must be recognized and converted into text, this approach initially requires more time than keyword spotting. But on the other hand, transcription makes the entire audio recording available as text so that any word can be found via full-text search.

The result (text) is entered in the INSPIRATION*neo* database via an XML interface.

An advantage of full-text searches is that the search terms can be displayed in context. This excludes misunderstandings, e. g. in the event of ambiguities. The texts are available for additional analyses and can be transferred to other systems to do so.

There is no upper limit for the length of the audio, however it must have a minimum length of 200 milliseconds to be processed.

Full-text search with Solr offers the possibility to define two words which appear in certain proximity of each other.



For information about full-text search refer to the user manual INSPIRATION*neo* - Sessions module.

ASC recommends using stereo calls for transcription.

Transcription for mono calls is possible but not advisable.



As all call participants are merged in one track in mono calls, the results would be associated with one participant. The audio analysis engine is not able to properly separate cross-talk occurring when participants speak at the same time; as a result, the quality of the transcription decreases.

Keyword spotting

By means of keyword spotting, you can filter for certain topics or categorize the sessions. To this end, you compile all expressions and phrases (keywords) which describe a topic in an analysis list. The defined keywords will then be searched automatically in the sessions.

Since this approach is limited to detecting individual words and phrases, sessions can be searched quickly. Since you have to define the expressions which are supposed to be searched for in advance, this approach especially serves to identify already known topics which frequently come up again.



For further information about the configuration of the audio analysis software EML Transcription Server refer to the administration manual *Configuration speech analysis*.



Additional information about creating audio analysis jobs and about how to use them can be found in the user manual *Usage Audio Analysis module*.

3 Installation

Install the speech analysis software according to the manufacturer's instructions following in this chapter.



Please note, that the speech analysis software must be installed on its own server with a separate PostgreSQL database.



Please note that for new installations and updates of language packs a decoder version $\geq 1.5.3$ and an EML Transcription Server version $\geq 1.4.2$ is required.



For further information about the update of EML speech analysis software refer to the installation manual for system providers *Software updates*.

See also

 [EML Transcription Server Version Windows 2.3.pdf](#) [► 7]

EML Transcription Server

Installation Instructions (Windows)

Version 2.3

(9 March 2021)

EML Speech Technology GmbH (EML)
Berliner Straße 45
D-69120 Heidelberg

Contents

1.	Overview	4
1.1	Basic Architecture	4
1.2	Streaming Setup	5
1.3	Ports.....	5
1.4	Workflow	6
1.5	Best-Practice Setup	6
2.	Database & Transcription Server.....	7
2.1	System Requirements - mandatory.....	7
2.1.1	Operating System	7
2.1.2	Hardware	7
2.1.3	Firewall.....	7
2.1.4	Software Dependencies.....	7
2.2	Installation.....	7
2.2.1	Database: PostgreSQL	8
2.2.2	Transcription Server.....	8
2.3	Post-Installation Steps	8
2.3.1	Create Database.....	8
2.3.2	Transcription Server user.....	8
2.3.3	Configure Wildfly service.....	9
2.3.4	Configure Wildfly Port	9
2.3.5	Install Wildfly service.....	10
2.4	Verify Installation.....	10
2.5	Configuration.....	10
2.5.1	Update Configuration	10
2.6	Populate Model and Project databases.....	11
2.6.1	Deploy models via GUI	11
2.6.2	Create Project	12
2.7	Verify Configuration.....	13
3.	Decoder.....	14
3.1	System Requirements - mandatory.....	14
3.1.1	Operating System	14
3.1.2	Hardware	14
3.1.3	Firewall.....	14
3.1.4	Software Dependencies.....	14

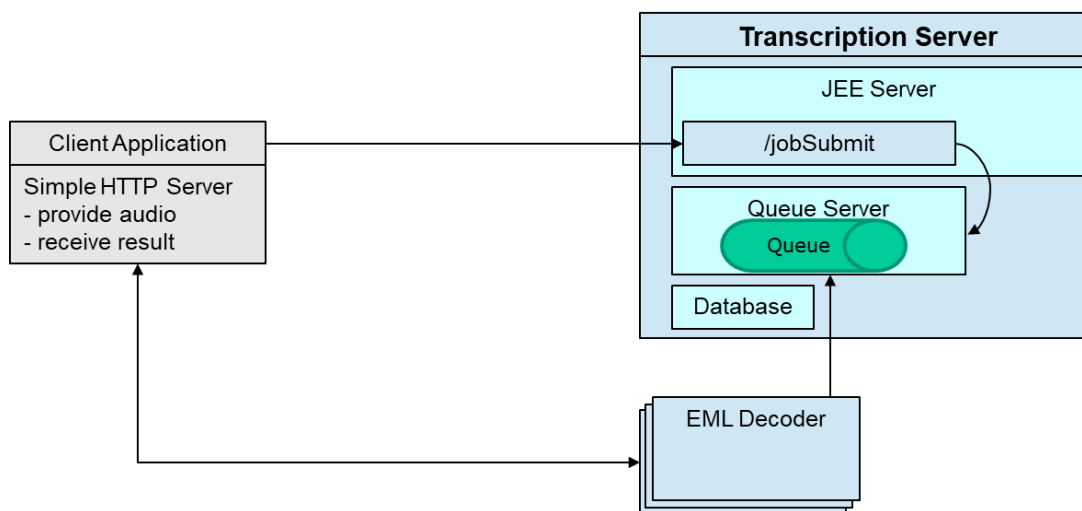
3.2	Installation.....	14
3.3	Configuration.....	15
3.4	Verify Installation & Configuration	15
4.	Streaming Service	16
4.1	Verify Installation.....	16
4.2	Configuration.....	16
4.3	Verify Configuration.....	16
5.	G2P	17
5.1	Verify Configuration.....	17
6.	Troubleshooting.....	18
6.1	EML Decoder Log	18
6.1.1	Log-Level	18
6.2	EML Transcription Server Log.....	18

1. Overview

The EML Transcription Server consists of the following components:

- *Decoder*: The decoder transcribes audio files from a queue server using a model.
- *Transcription Server*: The transcription server puts jobs into the queue the decoder listens to. It also provides a monitoring user interface for the queue server and manages the available models. Clients need to supply an audio URL from which the decoder can download the audio file and a second URL to which the decoder can send the results.
In order to be able to update a model independently of a specific client without the need to change the client, the transcription server keeps track of so called *projects*. A project is simply a mapping from a “project name” to a specific model. The client only provides the project name in the request to the transcription server, and the transcription server will translate it into a proper model name which can be used by the decoder.
- *Additional: Streaming Service*: The streaming server is used to establish a streaming transport of the audio and the result.
The streaming service introduces to notion of a language identifier (e.g. “en”) which is mapped in the streaming service to a project name of the transcription service, thus allowing changing the model or project independently of the clients (i.e. without needing to update the clients). Besides a streaming interface, the streaming service offers an easy to use HTTP interface to submit jobs (it allows to use a simple HTTP POST request to start transcription of an audio file).

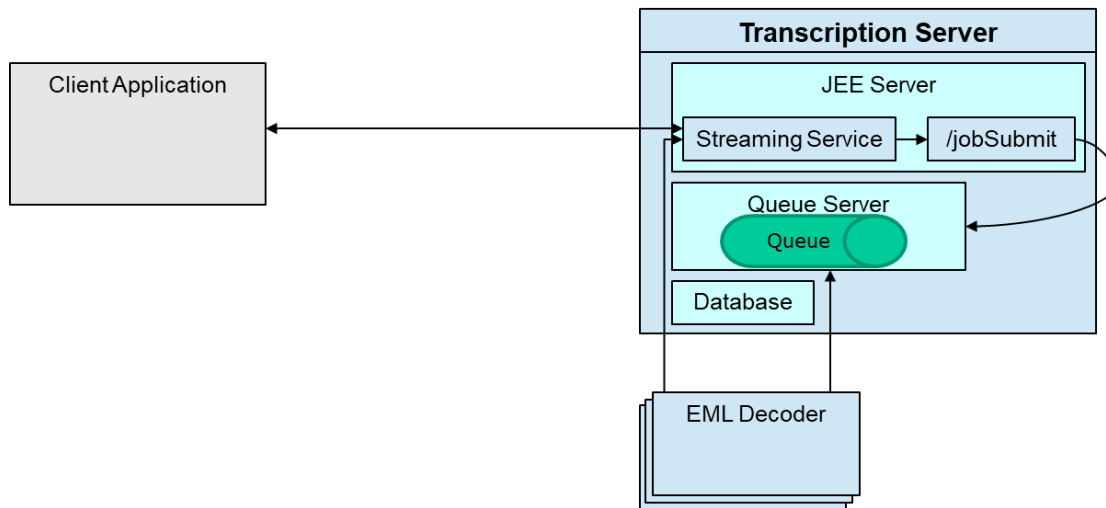
1.1 Basic Architecture



The basic architecture consists of the Transcription Server, the Decoder and a Client application. The Client sends a job description document to the transcription server. The job is put into a queue, where it is picked up by a decoder. The decoder requests the audio from the client, transcribes it and sends the result to the client. In the above picture, each of the orange boxes represents a different host, although it is possible to all of the components on one machine (given that the system requirements are met).

There can also be run multiple decoders which connect to the same queue to scale vertically. Multiple decoders can be run on one or different machines

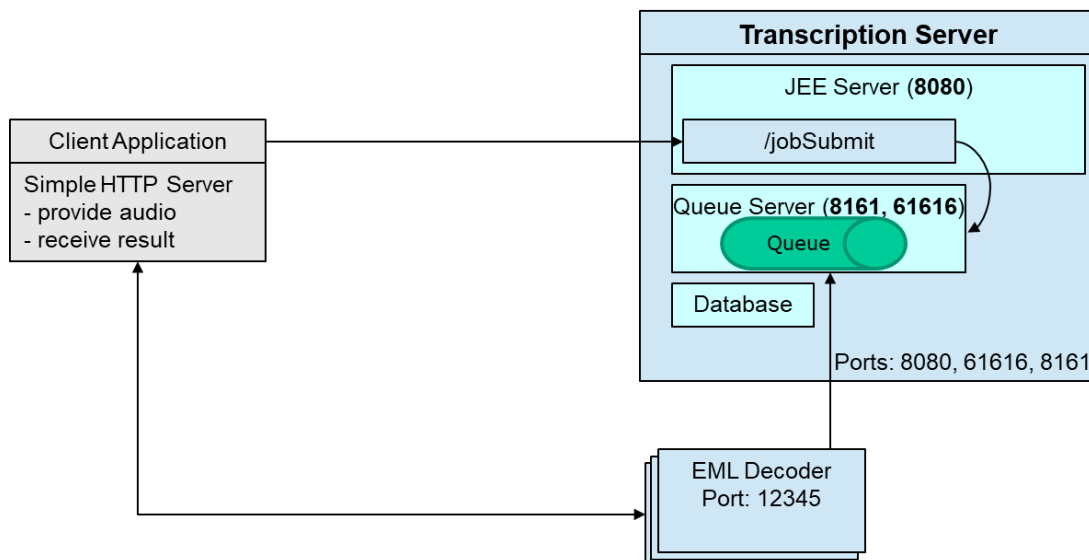
1.2 Streaming Setup



The client in this case holds a bi-directional connection to the streaming service, which in turn will post a corresponding job to the transcription server and will receive the results from the decoder. It can process the results and transmit only the requested information to the client, minimizing the amount of transferred data.

Since the streaming service is just an addition to the basic setup, both setups can be used in parallel.

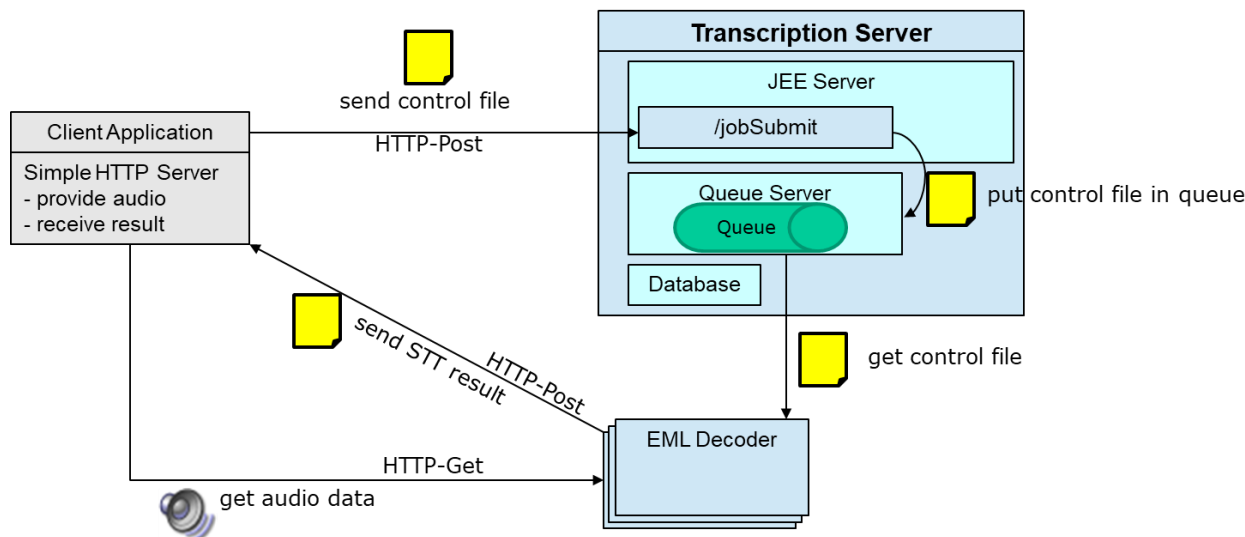
1.3 Ports



The Transcription Server is starting the services on ports: 8080/TCP, 61616/TCP and 8161/TCP. The Transcription Decoder will listen to port 12345/TCP when management console is activated (active by default).

Note: All ports used in JEE Server, Queue and Decoder are configurable!

1.4 Workflow



An application that uses EML Speech Transcription Server sends the control-file to Transcription Server with an HTTP POST request. The control file is fetched by one of decoders which are subscribed to the queue. In the control file is a pointer to an URI where the audio data can be downloaded. The decoding is then done on a local copy of the audio data.

The control-file is enriched with the result of the speech-to-text processing and the resulting file is sent back to the application using a call-back URI which is provided in the control-file. After the successful processing and the transfer of the result the processing is acknowledged to the queue.

1.5 Best-Practice Setup

For large scale setups we recommend installing the EML Transcription Server and the Queuing Server on one host and out the decoders on separate hosts. For smaller setups, where only a few channels are needed the EML Transcription Server, the Queuing Server and the decoders may run on the same host.

2. Database & Transcription Server

The EML Transcription Server is the service responsible for the automatic transcription of audio. It is a highly scalable JEE Application running in a JEE application server (e.g. JBoss Wildfly). It is responsible for the automatic model deployment to the registered EML Transcription Decoders as well as the logging of the job status.

As part of the EML Transcription Server the EML Transcription Server Monitor is deployed. This is a browser based graphical management tool.

2.1 System Requirements - mandatory

Note: On Software Dependencies and Operating Systems we expect 64bit versions.

The Transcription Server comes with the following system requirements:

2.1.1 Operating System

- **Windows Server 2012R2**
No special requirements
- **Windows Server 2016**
No special requirements
- **Windows Server 2019**
No special requirements

2.1.2 Hardware

- *CPU*: No special requirements
- *RAM*: 16GB
- *Disk*: 10GB + 10 GB per language

2.1.3 Firewall

- *Mandatory open:*
 - 8080/TCP (*JEE Application server e.g. Wildfly*)
 - 61616/TCP (*ActiveMQ OpenWire protocol*)
 - 8161/TCP (*ActiveMQ admin and REST interface*)

2.1.4 Software Dependencies

- **Java 8 JRE** (from Vendors like Oracle or AdoptOpenJDK or Amazon Corretto or Azul Zulu)

The environment variable JAVA_HOME is required. You may need to add a new entry in environment variables when the java installer will not create it.

- Name: JAVA_HOME
- Value: the directory where the Java software is located, for example
C:\Progra~1\Java\jre1.8.0_181

2.2 Installation

The EML Transcription Server will install several components: Apache ActiveMQ, Wildfly and EML Transcription Server JEE Applications.

2.2.1 Database: PostgreSQL

We strongly recommend to use PostgreSQL 9.5 or later (<https://www.postgresql.org/download/windows/>)

You can skip the database installation when there is already a proper PostgreSQL Version installed on your system.

Execute the installer and run through the wizard. Define postgres password (required also in a further configuration step). The other defaults are OK. You might want to change them, depending on your preferences. Skip the stackbuilder Wizard.

2.2.2 Transcription Server

To start installation issue double click:

EMLTranscriptionServer<version>.msi

There is no manual interaction required during installation. The EML Transcription Server is installed into

%ProgramFiles%\EML\TranscriptionServer

Other data like the model repository will be written into

%ProgramData%\EML\TranscriptionServer

ActiveMQ which is part of EMLTranscriptionServer.msi will be automatically installed as *Windows Service*. *Wildfly* will be installed as a Windows Service in a Post-Installation step.

2.3 Post-Installation Steps

2.3.1 Create Database

Start the PostgreSQL administration application "pgAdmin3" and double click on PostgreSQL 9.5.x (localhost:5432) and enter your password in the popup dialog. Right click on the node "Databases" and create a new database and enter as database name: sttdb

2.3.1.1 Fill Database



Activate the new database sttdb and click on the magnifying glass "SQL"

The .ddl files are located at: %ProgramFiles%\EML\TranscriptionServer\ddl

- Open "ddl_postgres_9.sql" and copy the content into the pgAdmin3 SQL Editor and execute the command (F5).
- Open "ddl_streaming_postgres_9.sql" and copy the content into the pgAdmin3 SQL Editor and execute the command (F5).

2.3.2 Transcription Server user

Note: Please activate in FileExplorer "show hidden files and directories" to make C:\ProgramData visible

The Installer is creating a Transcription Server user. This user will be used to access /Transcription-WebMonitor and /webSocket

The username and raw password is saved in a file

%ProgramData%\EML\TranscriptionServer\wildfly\transcriptionserver.users.raw.txt

You need to modify

%ProgramData%\EML\TranscriptionServer\streaming_config\configuration.properties
and replace the placeholder value "secret" in line "queue.pass" with the raw password from transcription-server.users.raw.txt

```
queue.host=http://localhost:8080
segmenter.minsil=60
queue.user=admin
queue.pass=secret
vmt.default_source=Android
service.port=8080
service.host=localhost
audio.dir=C:/ProgramData/EML/TranscriptionServer/tmp_audios
audio.dir.name.strategy=org.eml.stt.storage.names.PerKeyPerDateFile
NameStrategy
lang=de
```

2.3.3 Configure Wildfly service

Modify

%ProgramFiles%\EML\TranscriptionServer\wildfly\bin\service\service_config.bat
and replace "secret" in line 35: "set EMLDBPW=secret" with the database password defined in 2.2.1

```
set "EMLJDBC=jdbc:postgresql://localhost:5432/sttdb"
set "EMLDBUSER=postgres"
set "EMLDBPW=secret"
set "EMLLOGLVL=INFO"
```

2.3.4 Configure Wildfly Port

By default, the Wildfly Service will start listening on port 8080.

To check if port is already used by another process issue in CMD:

(EN) netstat -aon | findstr /R 8080 | findstr /i "listening"

(DE) netstat -aon | findstr /R 8080 | findstr /i "abhören"

When there is no output everything is fine and no other application is listening to that port. On the other side, if there is an output like this:

TCP	0.0.0.0:8080	0.0.0.0:0	ABHÖREN	3656
-----	--------------	-----------	---------	------

That means that there is already an application running on that port and you should change the wildfly port.

To change the port used by Wildfly modify:

%ProgramFiles%\EML\TranscriptionServer\wildfly\standalone\configuration\standalone.xml

```
<socket-binding name="http" port="${jboss.http.port:8080}"/>
```

Note: If you change the port you need to replace in all the example URL's 8080 with the port you choose. Keep in mind to also change the port in further Configuration steps: 2.5.1 and 3.3

2.3.5 Install Wildfly service

Run

```
%ProgramFiles%\EML\TranscriptionServer\wildfly\bin\service\service_install.bat
```

and the Windows Service *Wildfly* will be installed.

Note: the startup takes up to 60 seconds

2.4 Verify Installation

You should check the Configuration. Wait at least 60 seconds after installing the Wildfly Service and then login into the Transcription Server Monitor (<http://x.x.x.x:8080/TranscriptionWebMonitor> by default) and go to "Config". If you experience "404 – Not found" please check the log file mentioned in 0

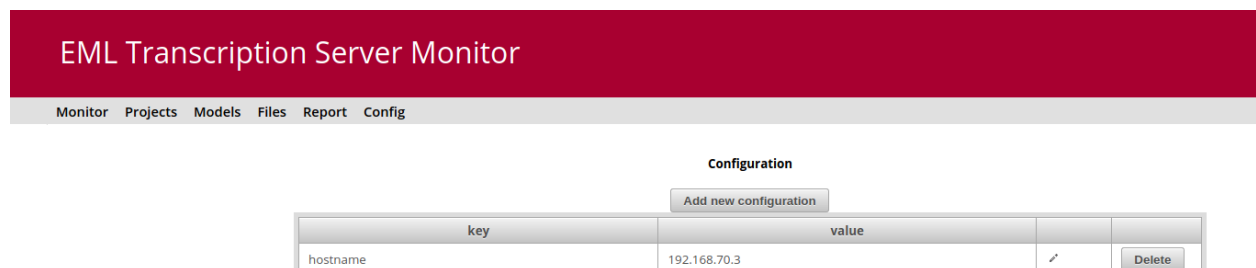
Note: The credentials to Transcription Server and other applications were saved during installation into

```
%ProgramData%\EML\TranscriptionServer\transcriptionserver.users.raw.txt
```

You can connect to the Active MQ admin interface (on <http://x.x.x.x:8161/> by default). Login credentials: (admin / admin) You should see at least one queue created eml-transcribe-stats with one listener. The queues are listed in "Manage ActiveMQ broker" -> "Queues"

2.5 Configuration

The installer will create an initial configuration with default values working well when all components (Transcription Server, Queue and Decoder) are running on same server [localhost]. When Decoder is running on a different server, it is important to update your hostname configuration to an IP or name from which the server is reachable.



Change "hostname" configuration to the IP of the server if it is a non-localhost setup

2.5.1 Update Configuration

The configuration is saved in the database and can be modified in "Config"-Tab in Transcription Server Monitor (on <http://x.x.x.x:8080/TranscriptionWebMonitor/> by default):

Key	Description	Default
-----	-------------	---------

hostname	The hostname or IP-address of the server the Transcription Server is running on and could be reached from the Transcription Decoders	<i>localhost</i>
port	The port on which the application server (Wildfly) is running	<i>8080</i>
repository.base	The directory on the server in which uploaded models will be stored.	<i>%ProgramData%\EML\TranscriptionServer\modelRepository</i>
sttweb.default-queue	The default queue to which jobs should be posted.	<i>eml-transcribe</i>
monitor.archiveFolder	The directory on the server in which generated reports will be stored.	<i>%ProgramData%\EML\TranscriptionServer\reportArchive</i>
sttweb.queue-server	The queue server identifier to use to post jobs. This will be appropriately set when using the <code>add-queue-server.sh</code> script.	<i>default</i>
sttweb.current-context	The web context in which the sttweb component is running. This will be automatically set. Do not change this value.	<i>/eml-stt</i>
monitor.current-context	The web context in which the EML Transcription Server Monitor component is running. This will be automatically set. Do not change this value.	<i>/TranscriptionWebMonitor</i>
sttweb.speaker-cache	The local cache directory where dynamic lexicon files and speaker adaptations will be saved.	<i>%TMP%</i>

2.6 Populate Model and Project databases

You need to populate the model and project database with entries in order to post jobs.

2.6.1 Deploy models via GUI

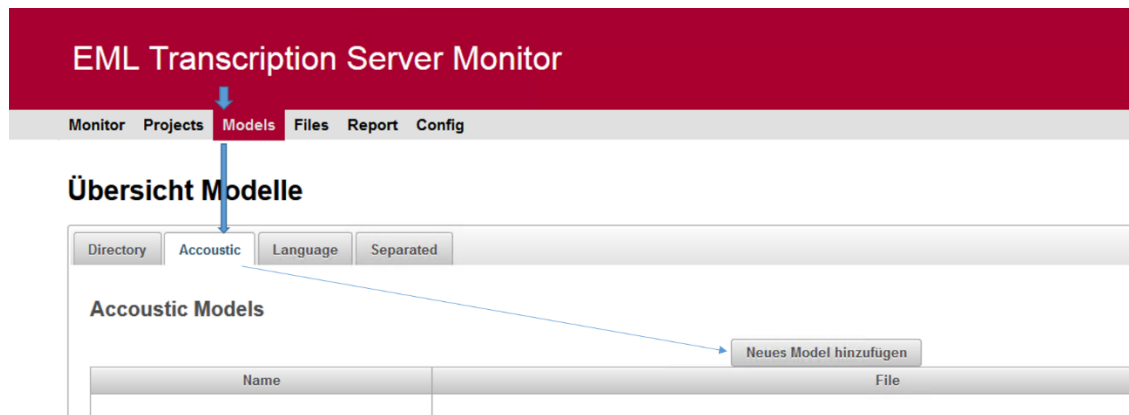
To upload and create entries manually, use the Transcription Server Monitor (on <http://x.x.x.x:8080/TranscriptionWebMonitor/> by default):

1. Go to the “Files” panel and upload your models files. Make sure to select the right type in the drop down menu. Usually the filename is an indicator of the type like “AM-...”, “LM-...”
Alternatively, if the model files are already present at the repository location set up above, you can simply import them.
2. Go to “Models”. Select the correct tab (“Directory” for complete models, “Acoustic” for Acoustic Models and “Language” for Language Models). Click the “Add new Model” button and fill out the form properly.

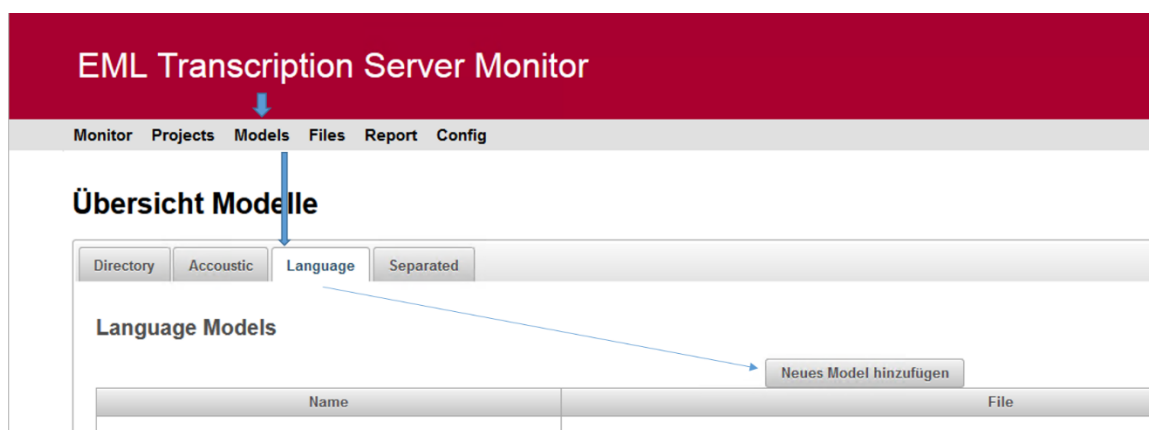
Example (Separated): You’ve got three files like

- i. “AM-bcn06.hbn.b.zip”
- ii. “LM-BCN_DeDE_10_161108_142740--2”

- iii. "IMAGES-bcn06.hbn.b_BCN_DeDE_10_161108_142740--2_cfg_stream.txt--1705.zip"



Create "Accoustic Model" (AM) entry



Create "Language Model" (LM) entry

IMAGES.zip does not need to be defined in "Models"! Now you can proceed with 2.6.2

Note that all of these operations also can be executed using the REST interface.

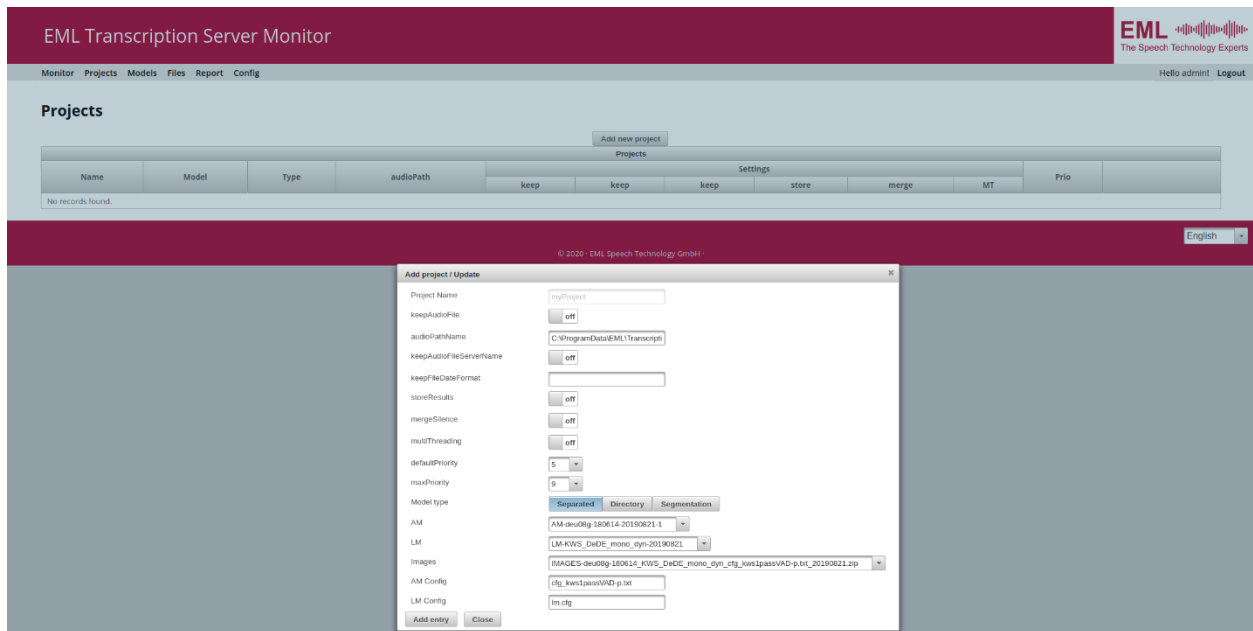
2.6.2 Create Project

Open Transcription Server Monitor and go to "Projects" in top menu navigation. Click the "Add new project" button. Select a project name and afterwards select the created model entries. There is a switch button "Separated | Directory" to get a list of either complete models (directory) or separated models (acoustic, language and images).

In the separated view there are two additional parameters: AM config and LM config:

The AM config is usually also listed in the filename of the images archive (e.g. if the image file is named IMAGES-bcn06.hbn.b_BCN_DeDE_10_161108_142740--2_cfg_stream.txt--1705.zip the AM config that has to be referenced is cfg_stream.txt).

The LM config is usually named lm.cfg.



Add new project dialog with separated model

2.7 Verify Configuration

After you have created the project entry, you can submit your first job to the Transcription Server. This job should appear in the proper queue (see the EML Transcription Server Monitor). Since no decoder is running yet, of course it will not get dispatched.

In order to post a job, the easiest way is to use the EML command-line tool `queue-files.bat` which is located in the `%ProgramFiles%\EML\TranscriptionServer\tools` directory.

```
queue-files.bat -if y.y.y.y -host x.x.x.x -queue <queue> -project <project> -
files <audio.wav>
```

Where `y.y.y.y` is the IP address of the computer on which the script runs and `x.x.x.x` is the host on which the EML Transcription Server runs. `<queue>` is usually "eml-transcribe" and `<project>` is the one created in 2.6.2

Since no decoder is yet running, the script will block indefinitely and you should abort it by pressing `Ctrl+C`.

If the job appears in the queue the installation is complete and working. Purge the queue in the EML Transcription Server Monitor before proceeding to set up the decoders.

3. Decoder

3.1 System Requirements - mandatory

The Decoder internally consists of several decoding threads, each acting as a full decoder instance. Since each thread loads a model into memory, the requirements depend on how many threads are configured to run.

Note: On Software Dependencies and Operating Systems we expect 64bit versions.

3.1.1 Operating System

- **Windows Server 2012R2**
No special requirements
- **Windows Server 2016**
No special requirements
- **Windows Server 2019**
No special requirements

3.1.2 Hardware

- **CPU:** One physical core per decoding thread
- **For Transcription:**
 - **RAM:** Depending on model, typically 4-8 GB per decoding thread
 - **Disk:** 20GB + 10 GB per language
- **For KWS**
 - **RAM:** 512MB per decoding thread
 - **Disk:** 5GB per language

3.1.3 Firewall

- **12345/TCP** (default management port used by EML Transcription Server Monitor)

3.1.4 Software Dependencies

- **Java 8 JRE** (from Vendors like Oracle or AdoptOpenJDK or Amazon Corretto or Azul Zulu)
- Microsoft Visual C++ Redistributable for Visual Studio 2015
- Transcription Server installed and configured

3.2 Installation

To start installation issue double click:

`EMLTranscriptionDecoder<version>.msi`

There is no interaction required during installation. The EML Transcription Decoder is installed into

`%ProgramFiles%\EML\emDecoder`

Other data like log files will be written into

`%ProgramData%\EML\emlDecoder`

EMLTranscriptionDecoder will be automatically installed as *Windows Service*.

3.3 Configuration

To configure the decoder, modify “decoder.properties” and make the changes required for your environment. Having a proper configuration is crucial. The default configuration assumes the EML Transcription Server has been installed on the same host as the decoder. If this is not the case, change the “defaultServer” parameter accordingly. If the default port of the Transcription Server was changed in a previous installation step you need change the “defaultServerPort”. The “defaultQueue” parameter sets the name of queue to which to connect. The queue will be created automatically if it is not existing in the queue server.

The “numInstances” parameter controls the number of decoding threads. The “modelRepositoryPath” parameter denotes the directory where the decoder will unpack the model files downloaded from the transcription server. Make sure the location exists and is writeable by the user the decoder is running as. Uncomment the “managementConsole” parameters to enable a REST service to query the state of the decoders and visualization in the EML Transcription Server Monitor.

To configure the logging system, modify “log4j2.xml”. The default settings are running on a minimum log on INFO level. Have a look at the log4j project on what this configuration file may contain. If you experience any issues on decoding, you can increase the log level to DEBUG.

Note: Changes on “decoder.properties” or “log4j2.xml” requires restarting the EMLTranscriptionDecoder Windows Service.

3.4 Verify Installation & Configuration

You can have a look at the log file

```
%ProgramData%\EML\emlDecoder\log\emlDecoder.log
```

For each instance you should see a line saying “Starting to process messages.” You can use the EML Transcription Server Monitor on the EML Transcription Server to check whether the Decoder is connected to the right queue. You should see for each decoding thread 1 consumer on the queue. This means the queue configuration on both ends is correct and working.

To check whether decoding is working, make sure that you have a project, which references to a model for which you have uploaded the model to the transcription server. If this is the case, submit a job for that project to the queue. You should see the job appearing in EML Transcription Server Monitor. The job will disappear from the queue *after* the job has been completely processed by the decoder. In the decoder’s log file, you should see several things happen: First of all, the message that a job was received, next details of the job will be printed (e.g. which model to use). After that, it is checked whether the model already exists in the model repository path, and if not it will be downloaded and extracted. If that is done the following workflow will take place:

1. Initialization of Engines: For each decoding there might be multiple engine processes that need to be spawned and initialized with the model. For future decodings with the same model, the already initialized engines will be re-used. So this is a one-time setup time.
2. Download of the audio: The audio is downloaded from the given URL.
3. Decoding
4. Sending the result(s): The results are posted to the given call back URL.

Please note, that the first few decodings will usually take longer (due to the initialization steps) then subsequent ones. Since the initialization time is a one-time setup, it is typically not included in real-time factor calculations.

4. Streaming Service

As the streaming service is part of the Transcription Server installation package no further installation is required. You can skip this part if it's not required to use streaming API.

4.1 Verify Installation

Point your browser to `http://x.x.x.x:8080/webSocket` and login with the Transcription Server credentials.

4.2 Configuration

After installation please check the Streaming Service configuration located at `%ProgramData%\EML\TranscriptionServer\streaming_config\configuration.properties`

After finishing on the configuration, create a new authentication key in menu "keys". After that, create a new language identifier. See the manual for the streaming service if you run into problems with that. Make sure you have setup the transcription server configuration items correctly before creating a language entry. Also make sure you have created at least one project in the transcription server. When creating a language id, you have to assign a project to it. Therefore, the streaming service requests a list of projects from the transcription server.

4.3 Verify Configuration

To verify that the configuration is working, use the REST interface with a WAV audio file with the proper sampling rate:

```
curl -H "Content-Type: application/octet-stream" --data-binary @<file>
"http://x.x.x.x:port/webSocket/rest/batch/<language-identifier>/<authentication-key>/?encoding=wav"
```

The result of this command should be the following actions:

- In the Transcription Server log file, you see a new request (look for RESTART messages)
- When the file has been transmitted, you will see a "Posting job to ..." message. If there is no exception, the message has been posted to the transcription server. The processing on the transcription server and on the decoder eventually starts (check those logs)
- In the Transcription Server log you will see more messages when the results arrive from the decoder.
- It is quickly processed and should be printed as result of the curl call on your command line.

After this step was successful, use our command line tools, to stream the audio file using the WebSocket connection. The processing steps detailed above should be the same.

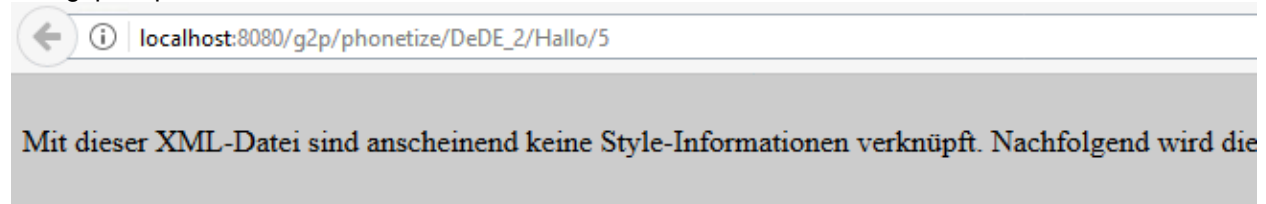
5. G2P

As the grapheme-to-phoneme (G2P) is part of the Transcription Server installation package no further installation is required. This is an optional tool to create pronunciations (e.g. used in KWS).

5.1 Verify Configuration

Point your browser to http://x.x.x.x:8080/g2p/phonetize/DeDE_2/Hallo/5 and wait a few seconds until you see the pronunciations. The first request may take 15-20 seconds until the engine is initialized.

The g2p output should look like



```
- <phonetizationResult>
  - <phonetizedWords wordClass="">
    <word>Hallo</word>
    - <phonetizations>
      <phonetization soundslike="Hallo" language="DeDE_2">h a l o:</phonetization>
      <phonetization soundslike="Hallo" language="DeDE_2">h a l O</phonetization>
      <phonetization soundslike="Hallo" language="DeDE_2">h E l o:</phonetization>
    </phonetizations>
  </phonetizedWords>
</phonetizationResult>
```

6. Troubleshooting

In case you have any problems you cannot solve yourself and you want to contact the support make sure to include the relevant parts of the EML Decoder Log and the EML Transcription Server Log.

6.1 EML Decoder Log

By default, the log file is located at

`%ProgramData%\EML\emlDecoder\log\`

6.1.1 Log-Level

By default, the Log-Level is set to INFO which only writes minimal information to the logs. If there is an issue with the Transcription Decoders please set the Log-Level to DEBUG, run your test again and send us the logs produced with DEBUG.

Edit `%ProgramFiles%\EML\emDecoder\log4j2.xml`

Replace:

```
<Logger name="de.emld" level="INFO" additivity="false">
<Logger name="org.eml" level="INFO" additivity="false">
```

By:

```
<Logger name="de.emld" level="DEBUG" additivity="false">
<Logger name="org.eml" level="DEBUG" additivity="false">
```

To apply the changes, install the Decoder service again by execute `service_uninst` and `sevice_inst`.

6.2 EML Transcription Server Log

By default, the log files are located at

`%ProgramFiles%\EML\TranscriptionServer\wildfly\standalone\log\`