

ABBYY

ABBYY® Timeline 6.0

Deployment Guide on Linux Systems

ABBYY® Timeline 6.0 - Deployment Guide on Linux Systems

© 2023 ABBYY. All rights reserved.

ABBYY® Timeline 6.0 © 2023 ABBYY Development Inc.

ABBYY is either registered trademark or trademark of ABBYY Software Ltd. in the United States and/or other countries.

All other product names, trademarks and registered trademarks are the property of their respective owners.

Information in this document is subject to change without notice and does not bear any commitment on the part of ABBYY.

The software described in this document is supplied under a license agreement. The software may only be used or copied in strict accordance with the terms of the agreement. It is a breach of the United States copyright law and international laws to copy the software onto any medium unless specifically allowed in the license agreement or nondisclosure agreements.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or other, for any purpose, without the express written permission of ABBYY.

Table of Contents

About This Document	4
Introducing ABBYY Timeline	5
System Requirements and Prerequisites	6
Installation, Removal and Upgrade	9
Installing Timeline	9
Performing Health Check	15
Upgrading from Previous Versions	17
Uninstalling Timeline	19
Timeline 5.3.0 patch installation	20
About Timeline Licenses	22
Updating a License	22
Administering	23
Configuring Timeline Using Environment File	23
HTTPS Configuration with SSL	30
Network Connection Settings	32
Background Upload of Zipped CSV Files to Timeline	33
Log Rotation	34
Known issues	36
Technical Support	40

About This Document

This deployment guide is intended for system administrators and engineers. It includes instructions for installation and configuration of ABBYY Timeline 6.0 on Linux.

Introducing ABBYY Timeline

ABBYY Timeline is a Process Intelligence platform featuring advanced process discovery, analysis, monitoring, simulation, and task mining. It uses the latest artificial intelligence (AI) to enable enterprises to automatically build an interactive digital twin of business processes that reveals inefficiencies and process bottlenecks. With this insight, they can apply automation where it will have the greatest impact and predict future outcomes.

ABBYY Timeline employs an exciting new patent-pending approach to process intelligence called Timeline Analysis which allows users to load events from a variety of systems and in different formats which it then automatically organizes into its corresponding process instances and allows them to be analyzed with a variety of visualization, discovery and query techniques. ABBYY Timeline accepts event data from any number of systems of record and automatically reconstructs the underlying business process logic behind the data. A variety of pre-built analyses are ready to quantify your process performance, identify your process execution issues and perform root cause analysis. The ABBYY Timeline platform also supports operational monitoring through its continuous assessment of new event data to determine if any adverse conditions occur and can immediately notify you or other business operations personnel so you can act.

The ABBYY Timeline engine consumes data from a variety of sources to detect and present detailed views of your business processes. This is often the same exact data being used today for other simpler analyses. This new insight is delivered via a variety of new process and timeline visualization tools developed to not only make these new insights easier to understand but also to allow users to manipulate the information to gain a deeper understanding of those processes. Users armed with this insight have concrete facts on which to take actions to improve operational efficiency by promoting clearly superior best practices and eliminating costly inefficiencies that previously went undetected.

Using advanced algorithms, ABBYY Timeline extracts and reads the time stamps used to record specific events along your processes. The software then visually models these time stamps in such a way that you can instantly identify deviations from an ideal process flow – to find the root cause of a problem that may be costing your business money.

ABBYY Timeline is aimed for use by anyone involved in business process improvements of any type of scale or nature.

System Requirements and Prerequisites

Operating system	<p>Red Hat Enterprise Linux 7.7</p> <p>Red Hat Enterprise Linux 8.5</p>
CPU	4 cores or more
RAM	16 GB or more
HDD	<p>512 GB or more</p> <p>Depends on the actual amount of data loaded into the application. Production environment may require more disk space, depending on the actual volume of data loaded into the application.</p>
Browser (to access the ABBYY Timeline website)	<ul style="list-style-type: none"> • Google Chrome 100 or later • Microsoft Edge 100 or later
Additional software	<p>Included in the installer:</p> <ul style="list-style-type: none"> • Redis 6.2.5 • NodeJS 16.15 • Python 3.8.10 • Nginx 1.20.1 <p>Downloaded from the Internet:</p> <p>Important. The installer automatically downloads and installs the following additional software from the Internet. If your machine is not connected to the Internet, the program will ask you to download it manually and prompt sources.</p> <ul style="list-style-type: none"> • container-selinux 2.119.2

	<ul style="list-style-type: none"> • Docker v.19.03.9 for Red Hat Enterprise Linux 7 v.20.10.9 for Red Hat Enterprise Linux 8 • Docker Compose 1.27.4 • Docker container 18.06.0-ce (with minideb docker-image) <p>Needs to be installed and configured before Timeline installation:</p> <ul style="list-style-type: none"> • SMTP Server ABBYY Timeline needs access to a running SMTP server to be able to send verification emails, notifications, invitations, and alerts, etc. • PostgreSQL 12 If you plan to use a remote database, make sure that you have a configured and running PostgreSQL 12 instance to which you will be connecting during Timeline installation. If you plan to use a local database, the PostgreSQL 12.6 is included in the installer, no additional actions are needed. • Twilio account You need to have a configured Twilio account if you want to enable SMS notifications feature in Timeline.
Other requirements	The target machine has to be connected to the Internet during the installation. If it is offline, you will have to download some additional software manually, in accordance with the installer prompts.

Scaling guidelines

The exact calculation of necessary hardware requires multiple parameters such as data volume and use patterns. However, the general guidelines could be defined as following:

- If the number of concurrent users is less than 10 and the data update frequency is one per day or less, a single server should be sufficient.
- For more users or more frequent data updates, a separate server for DBMS is recommended.

- For the fault-tolerant environment, use two identical servers and any standard load balancer.

Installation, Removal and Upgrade

This section provides instructions for installing ABBYY Timeline, updating, and removing it from your computers.

Installing Timeline

Before you begin

- Obtain the Timeline for Linux installer either from your ABBYY representative or ABBYY Technical Support by submitting a request on this [page](#).
- Verify all Linux [system requirements and prerequisites](#)^[6] before starting your Timeline installation.
- Ensure that any firewall installations are not blocking ports 80, 443 and 5432 or the ports you plan to setup for the Web Server and database. The installation will not work if firewall is blocking the ports that have been specified during the Timeline installation.
- If you intend to configure HTTPS, please visit the [HTTPS Configuration](#)^[30] section.

Procedure

1. Copy the Timeline installer to the local disk, typically to /tmp.f
2. Open the shell.

Important. You need root access to install Timeline. Without root, you won't have the necessary permissions to install it. Prefixing every command with **sudo** is cumbersome and it causes a problem with exported variables on the command line. If you are not using the host machine with the root user, start a new shell with root privileges using the command **sudo bash**.

3. Set the permissions of the **timeline-install-6.0.*.sh** file so that it is executable:
chmod +x /path/to/timeline-install-6.0.*.sh

4. Execute the Timeline install script:

path/to/timeline-install-6.0.*.sh

If the script is in the current directory, then you need to specify the dot (./) before the script file name:

./timeline-install-6.0.*.sh

Continue the installation when prompted. Installation workflow includes the following steps.

5. **Read and accept the license agreement**

Read the information in the End-User License Agreement. After reviewing the license agreement information, to indicate acceptance of the EULA, press the **Y** key. Any other input cancels the installation. After you accept the license agreement, installation proceeds.

6. **Check and install prerequisites**

- a. **PostgreSQL instance and database**

Timeline needs access to PostgreSQL 12. You can install PostgreSQL on a computer along with Timeline or a separate computer.

The program will ask you about using either local or remote database.

If you already installed PostgreSQL on another computer, select **Remote**. This option is useful if you install the program in a production environment where the host machine is accessible from outside the corporate network. If you prefer to keep Timeline and the database on the same machine, select **Local**.

- i. **Local**

If you want to use PostgreSQL on the host machine, select **Local**. The database will be set up by the installer and run inside the docker-container. This option is useful if you install the program for testing purposes or environments where the host machine is not accessible from outside the corporate network, and only the HTTP/HTTPS ports open.

During the installation process, the following PostgreSQL databases are created:

timeline

The database contains all information about users, their activity, and projects.

timeline-log

The database contains detailed records of Timeline events such as security, errors, and notifications.

timeline-000

The database contains information about user repositories.

ii. **Remote**

When you select this option, the program will ask you about establishing a secure SSL connection to the remote PostgreSQL database. Press **Y** if you plan to use SSL, or **N** to reject.

For several databases, located on different servers provisioned by different certificate authorities, set SSL root certificate paths for them separately.

Important. If your PostgreSQL is configured with SSL support and a root CA certificate file is used, you must provide the full path to the root CA certificate when configuring your connection settings. The certificate file will be copied to the **\$TIMELINE_INSTALLATION_DIR/db-ssl** folder.

To use this option, you must prepare **timeline**, **timeline-log**, and **timeline-000** databases in the remote PostgreSQL in advance:

1. Launch PostgreSQL
2. Create a user that can own database objects.
For example, TimelineUser.
3. Create the following databases owned by the user you created in the previous step:

timeline

The database contains all information about users, their activity, and projects.

timeline-log

The database contains detailed records of Timeline events such as security, errors, and notifications.

timeline-000

The database contains information about user repositories.

For setting up access to Timeline databases, you will be asked for the connection settings in one of the further installation steps.

b. **Docker and docker-compose**

Timeline runs in Docker containers so Docker and Docker-compose should be installed on the host machine. Docker is a Linux-based virtualization tool that helps to make complex applications more portable. You can install it manually or let the Timeline installer download and install it.

Important. In Timeline 6.0, the installer automatically downloads Docker and its dependencies from the Internet. If your machine is not connected to the Internet, the program will ask you to download it manually and prompt sources.

7. **Configure web server**

a. **HTTP and HTTPS port**

Specify the TCP/IP port for the Timeline website. Make sure that the specified port is not being used by any other application. By default, the application listens on port 80 for HTTP and port 443 for HTTPS. If both ports are defined, HTTP requests will be redirected to HTTPS.

You will be asked whether to use both HTTP and HTTPS, HTTP only, or HTTPS only. Navigate to the needed option using the arrow keys and press Enter to select it.

i. **HTTP and HTTPS ports**

Define ports for both options as described in further steps.

ii. **HTTP port (80)**

Press **Enter** to use the default 80 port or enter the port number.

iii. **HTTPS port (443)**

Press **Enter** to use the default 443 port or enter the port number.

If **HTTP and HTTPS** or **HTTPS** options are selected, after specifying the ports you will be prompted to specify the SSL certificate and key (and optionally a key password file). Provide the absolute path to these files. Then these files will be copied to the appropriate location (`$TIMELINE_INSTALLATION_DIR/nginx`) and the **ssl.conf** set up accordingly.

Then you will be asked whether to set the root certificate. Agree if you use self-signed certificates, otherwise this step can be skipped.

For preconditions and HTTPS related troubleshooting instructions, please see the [HTTPS Configuration with SSL](#) ^[30] section.

You can change ports configuration after installation. For more information, see [Configuring ABBYY Timeline Using .env File](#) ^[23] section.

b. **Base URL**

Enter the base URL that hosts Timeline or press **Enter** to use the default **https://127.0.0.1** one. It must be a public IP of the server or an external fully qualified URL. The lowercase pattern is recommended. Base URL also is used for links inside email messages sent by Timeline.

The Base URL must have the following syntax: **http[s]://hostname:port**
Important. Do not use extra spaces and forward slash '/' at the end of the base URL.

If you are using the default port (80 or 443), you do not need to add them to the base URL.

Examples:

The base URL of the HTTP endpoint, if a custom port is specified:

http://mytimeline.com:8080

The base URL of the HTTPS endpoint, if a custom port is specified:

https://mytimeline.com:30443

8. **Configure mail server**

Configure the SMTP server access to allow Timeline to send out emails in several features such as Alerting, User invitation, etc. You should provide general information to configure SMTP mail server and specify its security options. To decide which options you have to select, please refer to the documentation of your mail server. The server's basic settings are set during installation. You can change SMTP Mail Server configuration after installation. For more information, see [Configuring ABBYY Timeline Using .env File](#) ²³ section.

a. **Host**

Specify server name where SMTP mail server is installed.

b. **Port**

Enter SMTP mail server port number.

c. **Username** and **Password**

Enter SMTP mail server access credentials.

d. **E-Mail sender**

Enter the sender address that is used to fill the 'From' header field of e-mails.

e. **Use TLS? (Y/N)**

Enter **N** in case your SMTP server does not use TLS. It is a typical use case for mock, local mail services, for example, mailcatcher.

f. **Require TLS (Y/N)**

Enter **Y** if the initial connection should happen over an unencrypted connection and then the STARTTLS command should be used to upgrade to a secure connection. For example, Microsoft Exchange.

g. **Reject unauthorized (Y/N)**

Enter **Y** if your mail server uses SSL certificate issued by the Certification Authority (CA).

Enter **N** if your mail server uses an unauthorized, e.g., self-signed, SSL certificate.

9. **SMS service**

Configure Twilio SMS service to receive SMS notifications from Timeline containing verification codes, alert notifications, and error messages.

Important. Twilio account is required to configure and send SMS notifications.

a. **Do you want to use Twilio SMS service? (Y/N)**

Enter **N** in case you don't want SMS notifications. Enter **Y** to proceed with SMS notifications configuration.

b. **Twilio account SID**

Enter a String Identifier (SID), a unique key that will be used to identify your account when sending SMS notifications.

c. **Twilio Auth Token**

Enter your **Auth Token**, an access token that Timeline needs to connect to your **Twilio** account.

d. **Phone number**

Enter the phone number from your Twilio account. It will be the sender number.

10. **Configure admin user account**

Enter a valid e-mail address using an existing domain name that is configured to receive emails, for example, **user@domain.com**, and a password for it. This will be the first user of Timeline and the one that will have access to the ABBYY Timeline website, where other users can be administered.

The password you specified must contain only English letters and digits from 0 to 9.

Note. If you are upgrading Timeline, you are not prompted to enter admin user credentials because previous settings are maintained.

11. Database

This step appears if you choose Remote to connect to a remote PostgreSQL instance at the beginning of the installation process. The installer will ask for the settings for accessing the **Admin DB**, **Log DB**, **User DB** databases. For each of these databases, specify the connection parameters to **timeline**, **timeline-log**, and **timeline-000** databases located on the remote PostgreSQL.

a. PostgreSQL host

If you select **Connect to existing database** on the previous step, specify server name where PostgreSQL is installed. By default, **localhost** is used.

b. PostgreSQL port

Specify the TCP/IP port for PostgreSQL. By default, TCP/IP port **5432** is used. Make sure that it is not being used by any other application.

c. Database username

Provide the credentials of the PostgreSQL user who owns the Timeline databases. For example, TimelineUser.

d. Database name

- Enter **timeline** as the database name for the **Admin DB**.
- Enter **timeline-log** as the database name for the **Log DB**.
- Enter **timeline-** as the database name for the **User DB**. It is the prefix of the **timeline-000** database you created in the remote PostgreSQL.

12. [Perform Timeline health check](#) ¹⁵

Important. If you intend to configure HTTPS, you must set up SSL configuration after installation is complete. See [HTTPS Configuration with SSL](#) ³⁰ for details.

Performing Health Check

Check that Timeline is working properly by doing the following:

1. Make sure all docker containers are running on the host machine by using:

sudo docker ps -a

Note. You can ignore the status of **timeline_migrate_1** container which is used only to migrate databases and is not running after Timeline start.

2. Open a browser and enter **{timelineURL}:{port}** in the address bar, where:

{timelineUrl} is the Base URL you specified during the Timeline installation or the public IP address or the full name of the computer on which Timeline is installed.

{port} is the custom port assigned to the Timeline website during the installation process.

If you are using the default port (80 or 443), you do not need to add them to the **{timelineURL}**. By default, TCP/IP port 80 or 443 is used.

Example: http://mytimeline:8080 or https://mytimeline:30443

3. If the installation was carried out correctly, the Timeline website will open.
4. Login using the Timeline admin credentials you specified [during the installation process](#)  ¹⁴.

Upgrading from Previous Versions

If you own an older Timeline version, you can upgrade to ABBYY Timeline 6.0. Depending on the Red Hat Enterprise Linux version you use, there are the following preconditions:

- RHEL7
Upgrade from 5.2 is only possible within Red Hat Enterprise Linux 7 versions.
- RHEL8
Under Red Hat Enterprise Linux 8 the minimal version for upgrade is 5.3, since previous versions do not support this operating system.
- Clean installation on a separate server
If upgrade is not possible, or you need to do a clean installation on a new server, you can still connect a remote database used in the previous Timeline version by connecting to it in the **Connect to Database** step

This means that you can install a new version of ABBYY Timeline and your databases and previous settings will be maintained.

Procedure

To update your Timeline installation to the latest version, you have to execute the following steps:

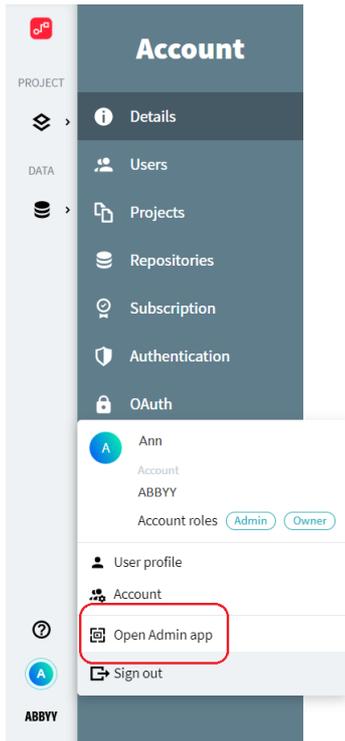
1. If you are using a local database, backup it and the **Storage** folder in the installation directory.
For detailed information, see <https://www.postgresql.org/docs/12/backup.html>.
2. Run the Timeline installation as described in the [Installing ABBYY Timeline](#) section and follow the instructions of the Installation Wizard. Press **Y** when the program will prompt you to upgrade.
3. When prompted **Import existing Timeline installation** select **opt/timeline** or **custom path** depending on where the program is already installed.
4. Select PostgreSQL location when prompted **Do you want ABBYY Timeline to use a local PostgreSQL instance or connect to a remote database?**
 - When upgrading from any supported version, database settings from any location are maintained.

- If you are doing a clean installation of a new version but want to use a remote database that was installed and configured with one of the previous Timeline versions, specify the parameters for connecting to this remote database at the Connect to Database step. For this scenario, only the databases created in PostgreSQL 12 are supported.
Note. If you choose the local instance option - a new database will be created.
- 5. If you plan to configure HTTPS with SSL specify HTTPS port when prompted.
- 6. To use existing databases, specify the values for connecting to them in the **Connect to Database** step.
Note. In case of remote database, please make sure that the correct values are entered. Incorrect user credentials (e.g. a typo in username) will cause a new database creation.
- 7. Complete the Timeline installation.
- 8. [Perform Timeline health check](#) 
- 9. After the installation process is complete, you need to rebuild all old Task Mining projects so that they are available to work in Timeline 6.0. For detailed instructions on rebuilding of the existing Task Mining project, see below.

How to Rebuild on the existing Task Mining project

1. Open a browser and enter **{timelineURL}:{port}** in the address bar, where:
{TimelineUrl} is the Base URL you specified during the Timeline installation or the public IP address or the full name of the computer on which Timeline is installed.
{port} is a custom port assigned to Timeline website during the installation process. If you are using the default port (80 or 443), you do not need to add them to the **{timelineURL}**. By default, TCP/IP port 80 or 443 is used.
Example: `http://mytimeline:8080` or `https://mytimeline:30443`
2. Login using the Timeline admin credentials you specified in the 'Admin User Account' step during the Timeline installation process.

3. Click your user avatar at the bottom of the left bar, then select **Open Admin app**.



4. In the **Admin** app, go to the **Project** tab and click **Rerun all task mining cutting**.
Important. Please do not refresh or close this browser page until the process is finished.

Uninstalling Timeline

To remove Timeline, execute the following commands, as root, to remove the docker containers and images, and other files. If you are not using the host machine with the root user, start a new shell with root privileges using the command **sudo bash**.

Important. Your local databases are stored in the **/opt/timeline** folder and will be deleted during the Timeline uninstallation process. If you are planning to use these databases in the future, backup them. For detailed information on how to backup your databases, see PostgreSQL documentation - <https://www.postgresql.org/docs/12/backup.html>.

1. user@host:~# docker kill \$(docker ps -q)
2. user@host:~# docker rm \$(docker ps -aq)
3. user@host:~# docker rmi \$(docker images -aq)
4. user@host:~# docker network rm timeline_network
5. user@host:~# rm /etc/systemd/system/timeline.service
6. user@host:~# rm -rf /opt/timeline

Timeline 5.3.0 patch installation

If you are using the Timeline 5.3.0 version and upload from an ODBC data source fails, you need to download the patch-file. [Click here to download](#).

This issue is fixed in later versions.

Prerequisites

- If the target machine (HOST) is not connected to the Internet, you will need an additional machine that has internet connection and docker installed (BUILDER). You need to have the **Dockerfile.odbcpatch** on the respective machine (HOST or BUILDER) depending on the HOST internet connection. All actions in the instruction are marked on which machine to perform them.
- Make sure you have root access on both machines.

Step 1 - Setup

1. HOST: Install the Timeline application.
2. HOST: Stop the timeline service:
`systemctl stop timeline`

Step 2 - Patch

1. Patch with internet connection on HOST.
 - a. HOST: Patch the **timeline/backend** image with the received dockerfile:
`cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t timeline/backend:latest .`
2. Patch without internet connection on HOST.
 - a. HOST: Export the **timeline/backend** image:
`docker save -o timeline-backend-latest.tar timeline/backend:latest`
 - b. Move the exported **timeline-backend-latest.tar** to the BUILDER machine.
 - c. BUILDER: Load the **timeline/backend** image:
`docker load --input <path_to_image_tar>/timeline-backend-latest.tar`
 - d. BUILDER: Patch the **timeline/backend** image with the received **dockerfile**:
`cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t timeline/backend:latest .`
 - e. BUILDER: Export the patched **timeline/backend** image:
`docker save -o timeline-backend-latest-patched.tar timeline/backend:latest`

f. Move the exported **timeline-backend-latest-patched.tar** to the HOST machine

g. HOST: Load the patched **timeline/backend** image:

```
docker load --input <path_to_image_tar>/timeline-backend-latest-patched.tar
```

Step 3 - Clean-up and start Timeline service

Perform these actions on the machine you have used for patch installation.

1. (Optional) HOST/BUILDER: Remove the exported image files

2. (Optional) HOST/BUILDER: Find the old **timeline/backend** image (repository: `timeline/backend`, tag: `<none>`):

```
docker image ls
```

3. (Optional) HOST/BUILDER: Delete the old **timeline/backend** image by id:

```
docker image rm <image_id_of_old_backend_image>
```

4. HOST: Restart timeline service:

```
systemctl start timeline
```

About Timeline Licenses

When installing ABBYY Timeline, the license copies to the computer on which Timeline is being installed. Your license determines the Timeline subscription plan. You need to purchase a license update in order to add support for new features, extend the license period, the maximum number of users you can add to your Timeline account, or change other license parameters.

To get more information about subscription plans or purchase a license, contact your local ABBYY sales representative or submit a request to the ABBYY technical support service on this [page](#).

Updating a License

After renewing your Timeline license, you need to update the license file on your host machine. To do it, proceed with the following steps:

1. Start the shell with root privileges using the `sudo bash` command.
2. Go to the `/opt/timeline/license` folder.
3. Backup the existing `timelinepi.lic` file.
4. Replace the `timelinepi.lic` file with the new one.
5. Restart the timeline service:

```
systemctl restart timeline
```

Administering

Chapter contents

- [Configuring Timeline Using Environment File](#) ²³
- [HTTPS Configuration with SSL](#) ³⁰
- [Network Connection Settings](#) ³²
- [Background Upload of Zipped CSV Files to Timeline](#) ³³
- [Log Rotation](#) ³⁴

Configuring Timeline Using Environment File

The Timeline settings may be configured after the installation.

At the system hosting Timeline, open the **opt/timeline/.env** file in any editor and set the following environment variables:

Parameters	Information	
Database connection settings		
ADMIN_DATABASE_URL	Description	Configures access to timeline , timeline-log and timeline-000 databases.
LOG_DATABASE_URL		
USER_DATABASE_URL_PREFIX	Format	<ul style="list-style-type: none"> • timeline and timeline-log database URLs have the following format: <pre>postgres://<username>:<password>@<IP address or postgres hostname>:<Port>/<Database name: timeline or timeline-log></pre> • timeline-000 database has a similar format with the difference that the last part defining the database should not contain the number '000': <pre>postgres://<username>:<password>@<IP address or postgres hostname>:<Port>/<Database name prefix: timeline-></pre>

Parameters	Information	
		<p><IP address or postgres hostname> should be the machine's IP or public name so it can be accessed from Docker containers.</p> <p>By default, PostgreSQL uses port 5432. Make sure that the configured port is not being used by any other application and is open on the firewall. If using the default PostgreSQL port, it can be done by:</p> <p>firewall-cmd --add-service=postgresql and firewall-cmd --runtime-to-permanent</p>
	Example	<pre>ADMIN_DATABASE_URL=postgres://trx:x@172.18.0.1:5432/timeline LOG_DATABASE_URL=postgres://trx:x@172.18.0.1:5432/timeline-log USER_DATABASE_URL_PREFIX=postgres://trx:x@172.18.0.1:5432/timeline-</pre>
Web server configuration		
PROXY_PORT PROXY_SSL_PORT	Description	<p>Specifies the ports configuration available for the application on the host machine.</p> <p>By default, the application listens on port 80 for HTTP and port 443 for HTTPS. If both ports are defined, HTTP requests will be redirected to HTTPS.</p> <p>For details on SSL configuration, see section 'HTTPS configuration'³⁰.</p> <p>Important.</p> <ul style="list-style-type: none"> • Make sure that the configured ports are open on the firewall and not being used by any other application.

Parameters	Information	
		<ul style="list-style-type: none"> If you install the application in a production environment, it is strongly recommended to use HTTPS and highly discouraged HTTP.
	Format	<p><i>PROXY_PORT</i>=<HTTP port></p> <p><i>PROXY_SSL_PORT</i>=<HTTPS port></p> <p>0 (zero) means the port is disabled.</p>
	Example	<p><i>PROXY_PORT</i>=0</p> <p><i>PROXY_SSL_PORT</i>=443</p>
BASE_URL	Description	<p>Specifies the Base URL that hosts Timeline. The hostname should include the port number if it is not the default and the protocol (http/https) of the server where the application is going to run.</p> <p>The BASE_URL variable is used for links inside email messages sent by Timeline.</p>
	Format	<p><i>BASE_URL</i>={protocol}://hostname[:port]</p>
	Example	<p><i>BASE_URL</i>=http://10.15.61.165 (if use HTTP)</p> <p><i>BASE_URL</i>=https://mytimeline.com (if use HTTPS)</p>
Mail server configuration		
MAIL_SERVER_HOST MAIL_SERVER_PORT MAIL_SERVER_USERNAME MAIL_SERVER_PASSWORD MAIL_SERVER_TLS_CONNECTION MAIL_SERVER_REQUIRE_TLS	Description	<p>Specifies SMTP server access to allow Timeline to send out emails in several features such as Alerting, User invitation, etc.</p> <p>Provide SMTP mail server access credentials such as host, port, username, password, e-mail sender address, and mail server security options.</p>

Parameters	Information	
<p>MAIL_SERVER_REJECT_UNAUTHORIZED</p> <p>EMAIL_SENDER</p>		<ul style="list-style-type: none"> • MAIL_SERVER_USERNAME MAIL_SERVER_PASSWORD Keep these fields empty if the mail server requires no authentication. • MAIL_SERVER_TLS_CONNECTION MAIL_SERVER_TLS_CONNECTION=true Makes the app connect to the mail server using TLS right from the start. This is the most secure option. Unfortunately, not all mail servers support this. E.g., Exchange requires unencrypted connection, and then use the STARTTLS command to upgrade. In this case, use: MAIL_SERVER_TLS_CONNECTION=false and MAIL_SERVER_REQUIRE_TLS=true. • MAIL_SERVER_REQUIRE_TLS To enable/disable TLS set MAIL_SERVER_REQUIRE_TLS to true/false. • MAIL_SERVER_REJECT_UNAUTHORIZED Set MAIL_SERVER_REJECT_UNAUTHORIZED to false if your mail server uses a self-signed certificate. Default value is true. • EMAIL_SENDER EMAIL_SENDER is used to fill the 'From' header field of e-mails.
	Format	<p><i>MAIL_SERVER_HOST=<mail server IP address or hostname></i></p> <p><i>MAIL_SERVER_PORT=<mail server port></i></p> <p><i>MAIL_SERVER_USERNAME=<mail server username></i></p>

Parameters	Information	
		<p><i>MAIL_SERVER_PASSWORD=<mail server password></i></p> <p><i>MAIL_SERVER_TLS_CONNECTION=<true/false></i></p> <p><i>MAIL_SERVER_REQUIRE_TLS=<true/false></i></p> <p><i>MAIL_SERVER_REJECT_UNAUTHORIZED=<true/false></i></p> <p><i>EMAIL_SENDER=<mail sender e-mail></i></p>
	Example	<p><i>MAIL_SERVER_HOST=example.smtp.server.com</i></p> <p><i>MAIL_SERVER_PORT=465</i></p> <p><i>MAIL_SERVER_USERNAME=mail_user</i></p> <p><i>MAIL_SERVER_PASSWORD=mail_password</i></p> <p><i>MAIL_SERVER_TLS_CONNECTION=false</i></p> <p><i>MAIL_SERVER_REQUIRE_TLS=true</i></p> <p><i>MAIL_SERVER_REJECT_UNAUTHORIZED=false</i></p> <p><i>EMAIL_SENDER=timeline-support@example.com</i></p>
SMS service (Twilio account)		
<p>TWILIO_ACCOUNT_SID</p> <p>TWILIO_AUTH_TOKEN</p> <p>TWILIO_PHONE_NUMBER</p>	Description	<p>These properties appear if Twilio service is enabled upon installation.</p> <p>Specifies Twilio account access to allow Timeline to send SMS notifications containing verification codes, alert notifications, and error messages.</p> <p>Provide Twilio account credentials such as SID, Auth Token, sender phone number. All</p>

Parameters	Information	
		<p>fields are mandatory, for correct configuration all fields should be filled.</p> <ul style="list-style-type: none"> • TWILIO_ACCOUNT_SID Is used to identify your Twilio account and serves as its username. • TWILIO_AUTH_TOKEN Is an access token that Timeline needs to connect to your Twilio account. • TWILIO_PHONE_NUMBER Valid phone number from which SMS notifications are sent.
	Format	<p><i>TWILIO_ACCOUNT_SID=<your Twilio account SID></i></p> <p><i>TWILIO_AUTH_TOKEN=<Authorization token belonging to your Twilio account></i></p> <p><i>TWILIO_PHONE_NUMBER=<PHONE_NUMBER></i></p>
	Example	<p><i>TWILIO_ACCOUNT_SID=AC3f84d59206412725a03114dfb5163e33</i></p> <p><i>TWILIO_AUTH_TOKEN=ae356b78c7ch1293h123n2afe6a9</i></p> <p><i>TWILIO_PHONE_NUMBER=+121313141516</i></p>
Timeline folders		
LOGS NGINX_CONF DB_SSL STORAGE LICENSE PG_SSL_ROOT_CERT_ADMIN	Description	Specifies the locations of directories the app saves data to. Each of these should be directories on the host machine. If you specify relative paths, they will be relative to the installation directory.

Parameters	Information
<p>PG_SSL_ROOT_CERT_LOG</p> <p>PG_SSL_ROOT_CERT_USER</p>	<ul style="list-style-type: none"> <li data-bbox="831 282 1433 412"> <p>• LOGS</p> <p>All Timeline logs will be placed here. Default value: /opt/timeline/logs</p> <li data-bbox="831 454 1433 725"> <p>• NGINX_CONF</p> <p>This is a directory for SSL configuration and certificates. Default value: /opt/timeline/nginx For details on SSL configuration, see section 'HTTPS configuration'.</p> <li data-bbox="831 768 1433 1182"> <p>• DB_SSL</p> <p>This is a directory for a database certificate file. If your remote PostgreSQL is configured with SSL support and a CA Root certificate file is not presented, the certificate file must be copied to the host machine into this directory. Default value: /opt/timeline/db-ssl</p> <li data-bbox="831 1225 1433 1733"> <p>• PG_SSL_ROOT_CERT_ADMIN PG_SSL_ROOT_CERT_LOG PG_SSL_ROOT_CERT_USER</p> <p>These are the names of the database CA Root certificate files located in the DB_SSL folder. The file names are hashed to avoid name collision. If your remote databases are configured with SSL support and a CA Root certificate files are used, this root certificate files must be specified in these keys.</p> <li data-bbox="831 1776 1433 1995"> <p>• STORAGE</p> <p>This directory is used by different parts of the application to permanently or temporarily store data. Make sure that the directories are not world readable and that</p>

Parameters	Information	
		<p>they are backed up regularly. Default value: /opt/timeline/storage</p> <ul style="list-style-type: none"> LICENSE The path to the directory where the license file is located relative to the installation directory. Default value: /opt/timeline/license <p>By default, all directories are under the installation directory.</p>
	Example	<pre>LOGS=/opt/timeline/logs NGINX_CONF=/opt/timeline/nginx STORAGE_DIR=/opt/timeline/storage LICENSE=/opt/timeline/license PG_SSL_ROOT_CERT_ADMIN=bd469b5612d1807776be854922291ffd.pem PG_SSL_ROOT_CERT_LOG=bd469b5612d1807776be854922291ffd.pem PG_SSL_ROOT_CERT_USER=bd469b5612d1807776be854922291ffd.pem</pre>

HTTPS Configuration with SSL

HTTPS with SSL options is configured within the installation steps. If you are having issues with HTTPS configuration or you want to change its settings, you can use this article as a troubleshooting guide.

HTTPS configuration preconditions

The application uses NGINX proxy to deliver HTTP requests from the browsers to the backend services. This proxy is responsible for SSL termination too.

To configure HTTPS, you need SSL certificates for Timeline. You can choose one of the following options:

1. Use SSL certificate issued by the Certification Authority (CA).
This is the recommended approach for the application installation that is intended for a production environment. The connection to the server will be secure and users will not get any warnings from the browser.
2. Use a self-signed SSL certificate.
If you do not have a signed certificate or if you only require a certificate for testing purposes, use a self-signed SSL certificate. However, in this case users will get warnings from the web browser about the use of a self-signed certificate as the server will not be considered secure.
Note. For self-signed HTTPS certificates, the root certificate is required.

Important. If you install the program in a production environment, it is strongly recommended to use HTTPS and highly discouraged HTTP.

Procedure

1. Obtain an SSL certificate and a private key.
2. Run the Timeline installation and follow the Installation Wizard. To enable SSL between application and client specify HTTPS port and Base URL for HTTPS port in the `Web Server` step.
For more information see '[Installing Timeline](#)^[9]'.
3. After the Timeline installation process is complete, do the following:
 - a. Find the **ssl.conf.tpl** and **ssl.conf** files in the `$TIMELINE_INSTALLATION_DIR/nginx` folder and rename the **ssl.conf.tpl** file to **ssl.conf**. Alternatively, merge the **ssl.conf.tpl** file with **ssl.conf**, if you made any changes in the **ssl.conf** file for the previous Timeline version.
Note. These files are copied to the `$TIMELINE_INSTALLATION_DIR/nginx` folder during the upgrade process. The folder is specified in the [NGINX_CONF variable in .env](#)^[28]. The **ssl.conf.tpl** file stores the latest SSL configuration settings.
 - b. Copy your SSL certificate and private key files to the `$TIMELINE_INSTALLATION_DIR/nginx` folder.

- i. If your private key and certificate files are not named **cert.key** and **cert.pem**, respectively, you should change the **ssl_certificate** and **ssl_certificate_key** entries in **ssl.conf** accordingly.
 - ii. If you have a password file for the SSL key, uncomment the line `#ssl_password_file $TIMELINE_INSTALLATION_DIR/nginx/conf/pass.file`; in **ssl.conf**, and copy your **pass.file** to the `$TIMELINE_INSTALLATION_DIR/nginx` folder. If necessary, change the path to the folder you specified during the installation process.
 - iii. If intermediate certificates should be specified in addition to a primary certificate, they should be specified in the same **cert.pem** file in the following order: the primary certificate comes first, then the intermediate certificates.
- c. Open **.env** file and check the following environment variables:
- i. **PROXY_SSL_PORT**
Make sure the HTTPS port you want to use is specified in the **PROXY_SSL_PORT** variable.
Example: `PROXY_SSL_PORT=443`
 - ii. **BASE_URL**
Make sure the HTTPS protocol is specified in the **BASE_URL** variable.
Example: `BASE_URL=https://mytimeline.com`
4. Restart the Timeline application to apply all the changes:
`systemctl restart timeline`
5. [Perform a health check](#) ¹⁵.

Network Connection Settings

The table below lists the ports that are used by default to access Timeline. If you are using a software or hardware firewall, make sure that the exception settings for Timeline have been set up accordingly on the computer where it is installed.

If you reassign port numbers in [PROXY_PORT and/or PROXY_SSL_PORT variables in the .env file](#) ²⁴, you will need to make changes to the appropriate firewall rules that you are using.

Application name	Protocol type	Port	Traffic direction	Use
Timeline	TCP/IP	80 or the port specified during the installation (if use HTTP) 443 or the port specified during the installation (if use HTTPS) 80 and 443 or the ports specified during the installation (if use both HTTP and HTTPS)	Inbound	HTTP and HTTPS connections to the Timeline website.
PostgreSQL	TCP/IP	5432	Inbound	Connections to a PostgreSQL database server from the computer where Timeline is hosted.

Background Upload of Zipped CSV Files to Timeline

The background-upload feature involves a folder that is monitored for files copied there. Whenever a new ZIP file is detected in that folder, the application grabs it and interprets it as an uploaded archive. The folder is defined as STORAGE/sftp, where [STORAGE is the variable in the .env file](#) ²⁸.

The upload file can be copied to the specified folder by any means. It can be the target of an SFTP upload, or it can be an otherwise shared folder.

Log Rotation

The log file generated by the application can quickly increase in size, and if you want to make sure it doesn't take up too much disk space, you can introduce log rotation. Log rotation will periodically clear the old logs, thus preventing the log file from taking up all the disk space.

On most Linux systems, the **logrotate** command is already located at **/usr/sbin/logrotate**. The way to set up log rotation can differ based on the kind of Linux distribution you use, whether you set up the application as root or as a simple user, whether you placed the application in **/opt** or in **/home**, and the exact location where the logs are placed. You will find 2 typical use cases below:

A) When the log files are placed inside **/opt/timeline** or some similar place, and **docker** is executed as a root user:

1. Create a log rotation config file, for example at: **/etc/logrotate.d/timeline**

The file should contain the following:

```
/opt/timeline/logs/* {
    size 1G
    copytruncate
    rotate 1
}
```

The path should point to the log file generated by the `docker-compose up` command. This particular configuration would clear the log file when it exceeds the size of 1 MB and copy its original content to another file

called **/home/<USER>/timeline/logs/docker-compose.log.1**. The next time the log rotation runs and finds that the log size exceeds 1 MB again, it overrides the **docker-compose.log.1** with its new contents and clears the original log file.

2. This assumes that the Linux system already has a log rotation installed and registered as a cron job. You can verify this by checking that an **/etc/logrotate.conf** file exists, and it contains the line **include /etc/logrotate.d**, and also that there is a file called **/etc/cron.daily/logrotate** that runs the command **/usr/sbin/logrotate /etc/logrotate.conf**. Different Linux distributions might have these files arranged in a different way.

B) When the log files are placed inside /home/<USER>, and are written by a non-root user:

1. Create a log rotation config file, for example at: **/home/<USER>/logrotate.conf**

The file should contain the following:

```
/home/<USER>/timeline/logs/* {
    size 1G
    copytruncate
    rotate 1
}
```

The path should point to the log file generated by the docker-compose up command.

This configuration works the same way as described in the previous case.

2. Register a cron job to run the log rotation procedure once a day.

Run the following command to create a user-specific cron job:

crontab -e

This will open a text editor where you can register cron jobs by adding lines like the following:

```
0 * * * * /usr/sbin/logrotate /home/$USER/logrotate.conf --
state /home/$USER/logrotate-state.txt
```

This line would result in the **logrotate** command executed once per day using the previously defined log rotation configuration, and storing its state in **logrotate-state.txt** (this can be any file, and doesn't have to exist at the beginning)

Important. For log rotation to work correctly, you have to write the log files in append mode (in bash '>>' instead of just '>'), otherwise the log file cannot be cut, since the process would keep writing at its current offset at the location that used to be the end of the file, even after the file was cleared.

Known issues

Not enough space for Timeline installation

Upon Timeline installation, large docker images are loaded. If there is not enough space to load them, that might cause the installer to fail. The lack of RAM may cause it either.

It is recommended to check the available memory and space before proceeding with the installation steps. To do it:

1. Open the shell.
2. Execute the following command to check how much memory is available on the machine:

```
free -h
```

3. Execute the following command to check how much space is on the machine.

```
df -h
```

Important. If you have multiple partitions (like /var and /tmp on the image example below), those need free space too, in addition to the target one, where you are planning to install Timeline. It is recommended to have about 10 GB free on these partitions to ensure successful installation.

4. Check the Timeline system requirements to see if you have enough memory and space for the installation. To do it, please see [System Requirements and Prerequisites](#)^[6].

Output example:

```
[root@ip-10-180-10-25 bin]# free -h
              total        used         free       shared  buff/cache   available
Mem:           3.7G         179M         3.3G           16M           267M         3.3G
Swap:          0B           0B           0B
[root@ip-10-180-10-25 bin]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.9G   0 1.9G   0% /dev
tmpfs           1.9G   0 1.9G   0% /dev/shm
tmpfs           1.9G  17M 1.9G   1% /run
tmpfs           1.9G   0 1.9G   0% /sys/fs/cgroup
/dev/xvda2      50G   8.2G 42G  17% /
tmpfs           379M   0 379M   0% /run/user/0
[root@ip-10-180-10-25 bin]#
```

Upload from an ODBC data source fails in Timeline 5.3.0

If you are using the Timeline 5.3.0 version and upload from an ODBC data source fails, please download the patch-file to resolve this issue. For details and instructions see [Patch installation](#)^[20]. This issue is fixed in later versions.

Timeline Application not accessible outside the installed machine under Red Hat Enterprise Linux

This issue only occurs on host machines running **Red Hat Enterprise Linux** as its installation uses docker network and includes the OS built-in firewall.

It can be identified when the application is not reachable from outside the network/machine, but is reachable from inside and the following command returns an HTML response:

```
curl localhost
```

In this case, please recreate the Docker network:

1. Stop the Timeline service, to make sure it does not try to use Docker:

```
service timeline stop
```

2. Remove all containers using the command below, when asked for confirmation accept, select **yes**:

```
docker container prune
```

3. List Docker networks:

```
docker network ls
```

- a. Inspect the Docker network to get its gateway ip:

```
docker network inspect timeline_network
```

```

1 [root@ip-10-180-10-144 timeline]# docker network inspect timeline_network
2 [
3   {
4     "Name": "timeline_network",
5     "Id": "906315bbf37e1bd0a1d8d32865c6e52eab886d906614a82188696d4652269e57",
6     "Created": "2022-06-20T12:42:05.920017826Z",
7     "Scope": "local",
8     "Driver": "bridge",
9     "EnableIPv6": false,
10    "IPAM": {
11      "Driver": "default",
12      "Options": {},
13      "Config": [
14        {
15          "Subnet": "172.19.0.0/16",
16          "Gateway": "172.19.0.1"      # the IP address that we need
17        }
18      ]
19    },

```

4. Delete the network related to Timeline. Use the id of **timeline_network**:

```
docker network rm timeline_network
```

5. Make sure the **timeline_network** got deleted using command from step 3:

```
docker network ls
```

6. Stop the Docker service:

```
service docker stop
```

7. Stop the PostgreSQL service:

```
service postgresql-12 stop
```

Note. Perform this step if you are using a local database. Skip it if you are using a remote database.

8. Temporarily delete all firewall rules using this command:

```
iptables --flush
```

9. Restart the firewall:

```
service firewalld stop
service firewalld start
```

10. Start the Docker service:

```
service docker start
```

11. Once the Docker service is started, create a new Docker network:

```
docker network create timeline_network
```

12. Flush the IP tables one more time after restart:

```
iptables --flush
```

13. Inspect the network via Docker, the IP address of the gateway might change:

```
network inspect timeline_network
```

14. Copy the gateway IP address from the network and make sure that you have consistent values in the **opt/timeline/.env** file:

Check the values of the following variables and change them if needed:

ADMIN_DATABASE_URL

LOG_DATABASE_URL

USER_DATABASE_URL_PREFIX

15. Perform the following steps if you are using a local database. Skip these steps if you are using a remote database.

a. Copy the gateway IP address from the network and make sure that you have consistent values in the following files:

i. **pg_hba.conf**

Add an entry at the end to allow the connection:

```
host all all $DOCKER_GATEWAY_IP md5
```

Also add the IP mask. E.g.:

```
host all all 172.10.0.1/24 md5
```

Default path for PostgreSQL 12 conf files: **/var/lib/pgsql/12/data/**

ii. **postgresql.conf**

Change the listen address with the value of the new **DOCKER_GATEWAY_IP**.

Default path for PostgreSQL 12 conf files: **/var/lib/pgsql/12/data/**

b. Restart the PostgreSQL service:

```
service postgresql-12 start
```

i. If you didn't stop your local PostgreSQL database service earlier, restart the service for it to use the new configuration files:

```
service postgresql-12 stop  
then  
service postgresql-12 start
```

16. Make sure that Docker service is running:

```
service docker status
```

17. Start the Timeline service:

```
service timeline start
```

18. Make sure that all the containers are up and running (it can take some time):

```
docker container ls -a
```

19. Check that the Timeline application is available. The following command should return an HTML response:

```
curl localhost
```

20. Make sure that the Timeline application is available outside, by opening the application on the DNS/hostname of the machine.

Upload from an ODBC data source fails in Timeline 5.3.0

If you are using the Timeline 5.3.0 version and upload from an ODBC data source fails, please download the patch-file to resolve this issue. For details and instructions see [Patch installation](#)²⁰.

This issue is fixed in later versions.

Technical Support

Should you have any questions regarding the use of ABBYY Timeline, please submit a request to the ABBYY technical support service on this [page](#).