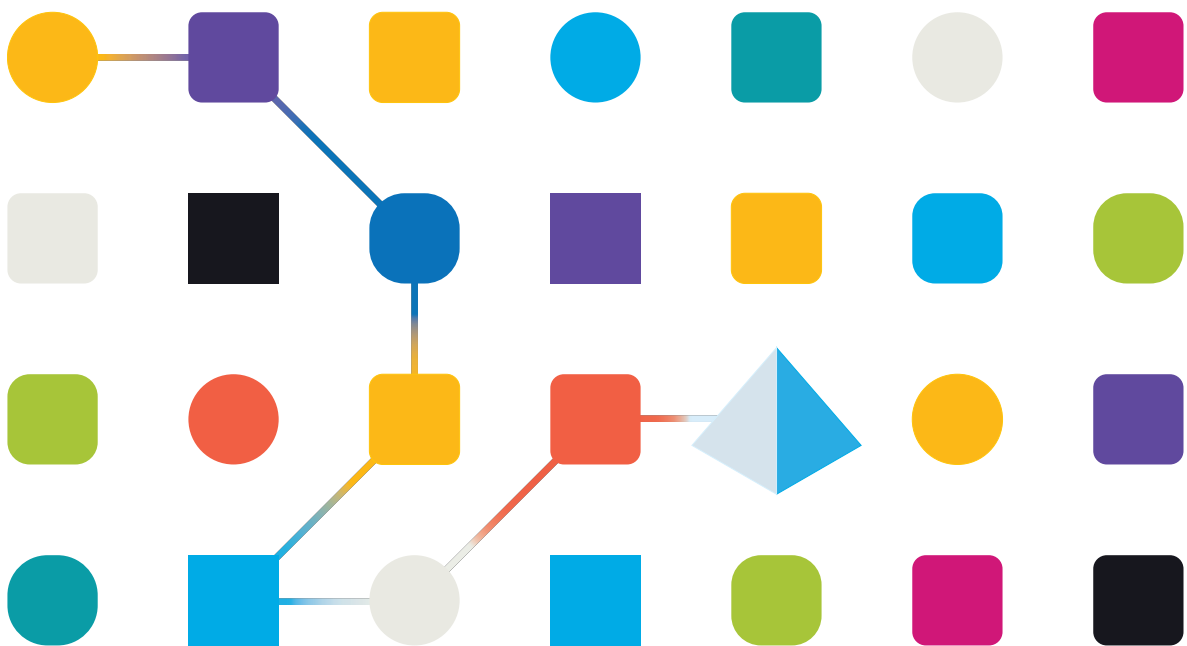




Process Intelligence 2.0

Installation Guide for Linux

Document Revision: 1.0



Trademarks and Copyright

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third-party without the written consent of an authorized Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

© Blue Prism Limited, 2001 – 2023

“Blue Prism”, the “Blue Prism” logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved.

All trademarks are hereby acknowledged and are used to the benefit of their respective owners. Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: www.blueprism.com

Contents

Installation overview	4
System requirements and prerequisites	5
Scaling guidelines	6
Install Process Intelligence	7
Uninstall Process Intelligence	12
Upgrade Process Intelligence	13
Rebuild existing Task Mining projects	14
Update a Process Intelligence license	15
Advanced configuration	16
Configure Process Intelligence using the environment file	16
HTTPS configuration	20
Network connection settings	22
Background upload	23
Log rotation	23
Patch installation	24
Troubleshooting	26
Not enough space for Process Intelligence installation	26
Process Intelligence is not accessible outside the installed machine when using Red Hat Enterprise Linux	26

Installation overview

This installation guide is intended for system administrators and engineers and includes instructions for installation and configuration of Blue Prism Process Intelligence on Linux.

- [System requirements](#)
- [Install Process Intelligence](#)
- [Uninstall Process Intelligence](#)
- [Upgrade Process Intelligence](#)
- [Advanced configuration](#)



The installation steps for the [Recording Service](#) and [Recorder](#) are the same as described in the Windows installation.

System requirements and prerequisites

Operating system	Red Hat Enterprise Linux 7.7 and 8.5
CPU	4 cores or more
RAM	16 GB or more
HDD	512 GB or more This depends on the actual amount of data loaded into the application. Production environments may require more disk space, depending on the actual volume of data loaded into the application.
Browser (to access the Blue Prism Process Intelligence 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>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• Docker<ul style="list-style-type: none">v.19.03.9 for Red Hat Enterprise Linux 7v.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>Need to be installed and configured before installing Process Intelligence:</p> <ul style="list-style-type: none">• SMTP Server<ul style="list-style-type: none">Process Intelligence needs access to a running SMTP server to be able to send verification emails, notifications, invitations, and alerts.• PostgreSQL 12<ul style="list-style-type: none">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 installation. If you plan to use a local database, the PostgreSQL 12.6 database instance is included in the installer and no additional actions are needed.• Twilio account<ul style="list-style-type: none">You must configure a Twilio account if you want to enable SMS notifications in Process Intelligence.
Other requirements	<p>The target machine must be connected to the Internet during the installation. If it is offline, you will have to download additional software manually. This will be prompted by the installer.</p>

Scaling guidelines

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

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

Install Process Intelligence

Before starting the installation, ensure the following:

- You have downloaded the Linux installation file for the version you require from the [Blue Prism Portal](#).
- All [Linux system requirements and prerequisites](#) have been met.
- Any firewall installations are not blocking ports **80**, **443** and **5432** or the ports you plan to set up for the web server and database. The installation will not work if a firewall is blocking the ports that have been specified during installation.
- If you intend to configure HTTPS, see [HTTPS configuration](#).

To install Process Intelligence on Linux, follow the steps below:

1. Copy the Process Intelligence installer to the local disk, typically to /tmp.
2. Open the shell.



You must have root access permissions to install Process Intelligence. Using `sudo` with each command could cause problems with exported variables on the command line, so if you are not using the host machine with the root user, it is recommended to start a new shell with root privileges using `sudo bash`.

3. Set the permissions of the **timeline-install-<version-number>.*.sh** file so that it is executable:

```
chmod +x /path/to/timeline-install-<version-number>0.*.sh
```

4. Execute the Process Intelligence install script:

```
path/to/timeline-install-<version-number>.*.sh
```

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

```
./timeline-install-<version-number>.*.sh
```

5. Continue the installation when prompted.
6. Read and accept the license agreement. Press **Y** to continue.

7. Install the prerequisites.

a. Install the PostgreSQL instance and database.

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

- **Local** – If you want to install PostgreSQL on the host machine or it is already installed there, select **Local**. 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.

You can install PostgreSQL using the native package manager on your system or let the Process Intelligence installer install it. During the installation process, the following PostgreSQL databases are created:

- **Local** – If you want to install PostgreSQL on the host machine or it is already installed there, select **Local**. 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.
 - **timeline** – The database contains all information about users, their activity, and projects.
 - **timeline-log** – The database contains detailed records of Process Intelligence events such as security, errors, and notifications.
 - **timeline-000** – The database contains information about user repositories.
- **Remote** – If you have 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 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 as follows:

- i. Launch PostgreSQL.
- ii. Create a user that can own database objects. For example, `TimelineUser`.
- iii. Create the following databases owned by the user you created at the previous step:
 - **timeline** – The database contains all information about users, their activity, and projects.
 - **timeline-log** – The database contains detailed records of Process Intelligence events such as security, errors, and notifications.
 - **timeline-000** – The database contains information about user repositories.

To set up access to Process Intelligence databases, you will be prompted for the connection settings in one of the following installation steps.



If your PostgreSQL database is configured with SSL support and a root CA certificate file is used, you must provide the 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.

b. Install Docker and docker-compose.

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

In Process Intelligence 2.0, 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.

8. Configure the web server.

a. Configure the HTTP and HTTPS ports.

Specify the TCP/IP port for the Process Intelligence website. Make sure that the specified port is used by any other application. By default, the application listens to port 80 for HTTP and port 443 for HTTPS. If both ports are defined, HTTP requests will be redirected to HTTPS. You can change the port configuration after installation. For more information, see [Configure Process Intelligence via environment file](#).

i. HTTP port (80)

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

ii. HTTPS port (443)

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

iii. HTTP and HTTPS ports


If the 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. These files will be copied to the appropriate location (\$TIMELINE_INSTALLATION_DIR/nginx) and the ssl.conf set up accordingly. You will then be prompted to set the root certificate. Agree if you use self-signed certificates, otherwise skip this step.

For more information, see [HTTPS configuration](#) and [Configure Process Intelligence using the environment file on page 16](#).

b. Configure the base URL.

Enter the base URL that hosts the Process Intelligence website, or press **Enter** to use the default https://127.0.0.1. This must be a public IP of the server or an external fully qualified URL. The lowercase pattern is recommended.

The base URL also is used for links inside email messages sent by Process Intelligence.

 The forward slash character '/' cannot be used at the end of the base URL.

The Base URL must have the following syntax: http[s]://hostname:port

If you are using the default port (80 or 443), you do not need to add this 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


9. Configure the mail server.

Configure access to the SMTP server to allow Process Intelligence to send out emails such as alerts or user invitations. You should provide general information to configure SMTP mail server and specify mail server security options. To decide which option you have to select, refer to the documentation of your mail server. Mail server basic settings are set during installation. You can change the SMTP Mail Server configuration after installation. For more information, see [Configure Process Intelligence via the environment file](#).

- a. **Host** – Specify the server name where the SMTP mail server is installed.
- b. **Port** – Enter the SMTP mail server port number.
- c. **Username and Password** – Enter the SMTP mail server access credentials.
- d. **Email sender** – Enter the email sender to be used in the **From** header field of emails.
- e. **Use TLS? (Y/N)** – Enter **N** if the SMTP server does not use Transport Layer Security (TLS). TLS is used 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 certificate, for example, a self-signed SSL certificate.

10. Configure the SMS service.

Configure the Twilio SMS service to receive SMS notifications from Process Intelligence including verification codes, alert notifications, and error messages.

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

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

Enter **N** if you don't want to receive SMS notifications. Enter **Y** to proceed with the SMS service configuration.

b. **Twilio account SID**

Enter your Twilio account string identifier. This is a unique key that will be used to identify your account when sending SMS notifications.

c. **Twilio auth token**


Enter your Twilio account authorization token, an access token that Process Intelligence needs to connect to your Twilio account.

d. **Phone number**

Enter the phone number from your Twilio account. This will be the sender number when sending SMS notifications.

11. Configure the admin user account.

Create administrator user? (Y/N) – Enter **Y** and a valid email address that is configured to receive emails (for example, user@domain.com), and a password. This will be the first user, and the one that will have access to the Process Intelligence website, where other users can be administered.

 When you upgrade Process Intelligence, you are not prompted to enter admin user credentials because previous settings are maintained.

12. Configure the database.

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

- a. **PostgreSQL host** – If you selected **Connect to existing database** at the previous step, specify the server name where PostgreSQL is installed. By default, localhost is used.
- b. **PostgreSQL port** – Specify the TCP/IP port for the 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 Process Intelligence 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. This is the prefix of the timeline-000 database you created in the remote PostgreSQL.

13. Perform an installation health check.

Verify that Process Intelligence is working correctly by doing the following:

- a. Make sure all docker containers are running on the host machine by entering:

```
sudo docker ps -a
```

You can ignore the status of the timeline_migrate_1 container. This is used only to migrate databases and is not running after Process Intelligence has started.

- b. Open a browser and enter **{URL}:{port}** in the address bar, where:
 - **{Uri}** is either the base URL you specified during the Process Intelligence installation, or the public IP address, or the full name of the computer on which Process Intelligence is installed.
 - **{port}** is the port assigned to the Process Intelligence website during the installation process. By default, TCP/IP port 80 is used.


Example: <http://myprocessintelligence:8080> or <https://myprocessintelligence:30443>

- c. If the installation was carried out correctly, the Process Intelligence website will launch.
- d. Log in using the Process Intelligence admin credentials you specified in [Configure the admin user account](#).

During the installation process, firewall exceptions are created, allowing interactions between components to take place inside a network. For default network connection settings, see [Network connection settings](#).

Uninstall Process Intelligence

To uninstall Process Intelligence, execute the following commands as root user, 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`.

 Your local databases are stored in the `/opt/timeline` folder and will be deleted during the Process Intelligence uninstallation process. If you plan to use these databases in the future, you must back them up. For more information, see the [PostgreSQL documentation](#).

Enter the following commands in order:

1. `user@host:~# docker-compose down`
2. `user@host:~#sudo systemctl restart docker`
3. `user@host:~# rm -rf timeline` (in Process Intelligence/Timeline directory)

Upgrade Process Intelligence

If you have installed a previous version of Process Intelligence, you can upgrade to a later version and your databases and previous settings will be maintained.

The prerequisites depend on the Red Hat Enterprise Linux version you use, as follows:

- **RHEL7** – Upgrade from Process Intelligence 1.0 is only possible when using Red Hat Enterprise Linux 7.
- **RHEL8** – Under Red Hat Enterprise Linux 8 the earliest version for upgrade is Process Intelligence 1.1, as previous versions do not support this operating system.
- **Clean installation on a separate server** – If an upgrade is not possible, or you need to do a fresh installation on a new server, you can still connect a remote database used in the previous Process Intelligence version by connecting to it during the installation process.

To update your Process Intelligence installation to the latest version:

1. If you are using a local database, back it up along with the Storage folder in the installation directory. For more details, see [Back up and restore the PostgreSQL databases](#).
2. Run the Process Intelligence installation as described in [Install Process Intelligence](#) and follow the instructions in the installation wizard.
3. When prompted to **Import existing Timeline installation**, select **opt/Timeline** or **custom path** depending on where the application is already installed.
4. Select the 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 of Process Intelligence, database settings from any location are maintained.
 - If you are carrying out a clean installation of a new version of Process Intelligence but want to use a remote database that was installed and configured with one of the previous versions, specify the parameters for connecting to this remote database when prompted during installation. For this scenario, only the databases created in PostgreSQL 12 are supported. If you choose the local instance option, a new database will be created.
5. If you plan to configure HTTPS with SSL (recommended), specify the HTTPS port when prompted.
6. If you plan to use an existing database, specify the existing values for connecting to it when prompted during installation.



If using a remote database, make sure that the correct values are entered. Incorrect user credentials will cause a new database creation.

7. Complete the Process Intelligence installation.
8. After the installation process is complete, configure HTTPS with SSL if necessary.

During the installation process the `ssl.conf.tpl` file is copied to the `$TIMELINE_INSTALLATION_DIR/nginx` folder. This file stores the SSL configuration settings. To set up SSL locate the `$TIMELINE_INSTALLATION_DIR/nginx` folder and do one of the following:

- Rename the **ssl.conf.tpl** file to **ssl.conf**.
- Alternatively, merge the **ssl.conf.tpl** file with **ssl.conf**. Use this method if you made any changes to the `ssl.conf` file for a previous version of Process Intelligence.

9. [Perform an installation health check](#).

Rebuild existing Task Mining projects

After the upgrade process is complete, you need to rebuild all your previous Task Mining projects so that they are available in Process Intelligence 2.0.

1. Open a browser and enter {URL}:{port} in the address bar, where:
 - {Url} is either the Base URL you specified during the Process Intelligence installation, or the public IP address or the full name of the machine on which Process Intelligence is installed.
 - {port} is a custom port assigned to the Process Intelligence website during the installation process. If you are using the default port (80 or 443), you do not need to add them to the {URL}. By default, TCP/IP port 80 or 443 is used.

Example: `http://myprocessintelligence:8080` or `https://processintelligence:30443`

2. Log in using the first admin account credentials you specified during the Process Intelligence installation process.
3. Click your user avatar in the navigation menu and select **Open Admin app**.
4. In the Admin app, navigate to the Project tab and click **Rerun all task mining cutting**.



Do not refresh or close this browser page until the process has finished.

Update a Process Intelligence license

 You must have system administrator privileges to update a Process Intelligence license.

To update your license:

1. On the computer on which Process Intelligence is installed, stop the timelinepi service using the Services snap-in or open Command Prompt as administrator and enter: `sc stop timelinepi`
2. Navigate to the Process Intelligence installation folder and open the license folder. The default is C:\Program Files\Blue Prism Process Intelligence powered by ABBYY Timeline\license.
3. Back up the existing timelinepi.lic file.
4. Replace the timelinepi.lic file with the new license file.
5. Start the timelinepi service using Services snap-in, or run Command Prompt as administrator and enter: `sc start timelinepi`


Advanced configuration

Configure Process Intelligence using the environment file

Process Intelligence settings can be configured after the installation by updating the environment file.

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

Parameters	Information	
Database connection settings ADMIN_DATABASE_URL LOG_DATABASE_URL USER_DATABASE_URL_PREFIX	Description	Configures access to the timeline , timeline-log and timeline-000 databases.
	Format	<ul style="list-style-type: none"> timeline and timeline-log database URLs has one of the following formats: postgres://<username>:<password><IP address or postgres hostname>:<Port>/<Database name: timeline or timeline-log> <ul style="list-style-type: none"> postgres://<username>:<password><IP address> postgres hostname>:<Port>/<Database name: timeline or timeline-log> timeline-000 database has a similar format with the difference that the last part defining the database should not contain the number '000': <ul style="list-style-type: none"> postgres://<username>:<password><IP address> postgres hostname>:<Port>/<Database name: timeline-> <p>IP address or postgres hostname must 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 used by any other application and is open on the firewall. If using the default PostgreSQL port, it can be done by using: 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>

Parameters	Information	
<p>Web server configuration</p> <p>PROXY_PORT</p> <p>PROXY_SSL_PORT</p>	<p>Description</p>	<p>Specifies the port 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 HTTPS configuration.</p> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p> Make sure that the configured ports are open on the firewall and not being used by any other application. If you install the application in a production environment, you are strongly recommended to use HTTPS and not HTTP.</p> </div>
	<p>Format</p>	<pre>PROXY_PORT=<HTTP port> PROXY_SSL_PORT=<HTTPS port></pre> <p>0 (zero) means that the port is disabled.</p>
	<p>Example</p>	<pre>PROXY_PORT=0 PROXY_SSL_PORT=443</pre>
<p>BASE_URL</p>	<p>Description</p>	<p>Specifies the base URL that hosts the Process Intelligence website. 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 Process Intelligence.</p>
	<p>Format</p>	<pre>BASE_URL={protocol}://hostname[:port]</pre>
	<p>Example</p>	<pre>BASE_URL=http://10.15.61.165 (if using HTTP) BASE_URL=https://mytimeline.com (if using HTTPS)</pre>

Parameters	Information	
<p>Mail server configuration</p> <p>MAIL_SERVER_HOST</p> <p>MAIL_SERVER_PORT</p> <p>MAIL_SERVER_USERNAME</p> <p>MAIL_SERVER_PASSWORD</p> <p>MAIL_SERVER_TLS_CONNECTION</p> <p>MAIL_SERVER_REQUIRE_TLS</p> <p>MAIL_SERVER_REJECT_UNAUTHORIZED</p> <p>EMAIL_SENDER</p>	<p>Description</p>	<p>Specifies SMTP server access to allow Process Intelligence to send out emails such as alert and user invitations.</p> <p>Provide SMTP mail server access credentials such as host, port, username, password, email sender address, and mail server security options.</p> <ul style="list-style-type: none"> • <code>MAIL_SERVER_TLS_CONNECTION</code> <code>MAIL_SERVER_TLS_CONNECTION=true</code> makes the app connect to the mail server using TLS from the start. This is the most secure option. However, not all mail servers support this. For example, Exchange requires unencrypted connection, and then uses the STARTTLS command to upgrade. In this case, use: <code>MAIL_SERVER_TLS_CONNECTION=false</code> and <code>MAIL_SERVER_REQUIRE_TLS=true</code>. • <code>MAIL_SERVER_REQUIRE_TLS</code> To enable/disable TLS, set <code>MAIL_SERVER_REQUIRE_TLS</code> to <code>true</code> or <code>false</code>. • <code>MAIL_SERVER_REJECT_UNAUTHORIZED</code> Set <code>MAIL_SERVER_REJECT_UNAUTHORIZED</code> to <code>false</code> if your mail server uses a self-signed certificate. The default value is <code>true</code>. • <code>EMAIL_SENDER</code> <code>EMAIL_SENDER</code> is used to fill the 'From' header field of e-mails.
	<p>Format</p>	<pre>MAIL_SERVER_HOST=<mail server IP address or hostname> MAIL_SERVER_PORT=<mail server port> MAIL_SERVER_USERNAME=<mail server username> MAIL_SERVER_PASSWORD=<mail server password> MAIL_SERVER_TLS_CONNECTION=<true/false> MAIL_SERVER_REQUIRE_TLS=<true/false> MAIL_SERVER_REJECT_UNAUTHORIZED=<true/false> EMAIL_SENDER=<mail sender e-mail></pre>
	<p>Example</p>	<pre>MAIL_SERVER_HOST=example.smtp.server.com MAIL_SERVER_PORT=465 MAIL_SERVER_USERNAME=mail_user MAIL_SERVER_PASSWORD=mail_password MAIL_SERVER_TLS_CONNECTION=false MAIL_SERVER_REQUIRE_TLS=true MAIL_SERVER_REJECT_UNAUTHORIZED=false EMAIL_SENDER=timeline-support@example.com</pre>

Parameters	Information	
<p>SMS service (Twilio account)</p> <p>TWILIO_ACCOUNT_SID</p> <p>TWILIO_AUTH_TOKEN</p> <p>TWILIO_PHONE_NUMBER</p>	Description	<p>These properties display if the Twilio service is enabled upon installation.</p> <p>Specifies Twilio account access to allow Process Intelligence to send SMS notifications containing verification codes, alert notifications, and error messages.</p> <p>Provide Twilio account credentials such as SID, access token, and sender phone number. All fields are mandatory.</p> <ul style="list-style-type: none"> • TWILIO_ACCOUNT_SID – Identifies your Twilio account and serves as its username. • TWILIO_AUTH_TOKEN – The access token that Process Intelligence needs to connect to your Twilio account. • TWILIO_PHONE_NUMBER – A valid phone number from which SMS notifications are sent.
	Format	<pre>TWILIO_ACCOUNT_SID=<your Twilio account SID> TWILIO_AUTH_TOKEN=<Authorization token belonging to your Twilio account> TWILIO_PHONE_NUMBER=<PHONE_NUMBER></pre>
	Example	<pre>TWILIO_ACCOUNT_SID=AC3f84d59206412725a03114dfb5163e33 TWILIO_AUTH_TOKEN=ae356b78c7ch1293h123n2afe6a9 TWILIO_PHONE_NUMBER=+121313141516</pre>


Parameters	Information	
<p>Process Intelligence folders</p> <p>LOGS</p> <p>NGINX_CONF</p> <p>DB_SSL</p> <p>PG_SSL_ROOT_CERT</p> <p>STORAGE</p> <p>LICENSE</p>	<p>Description</p>	<p>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.</p> <ul style="list-style-type: none"> <p>LOGS</p> <p>All Process Intelligence logs will be placed here.</p> <p>Default value: <code>/opt/timeline/logs</code></p> <p>NGINX_CONF</p> <p>This is a directory for SSL configuration and certificates.</p> <p>Default value: <code>/opt/timeline/nginx</code></p> <p>For details on SSL configuration, see HTTPS configuration.</p> <p>DB_SSL</p> <p>This is a directory for a database certificate file.</p> <p>If your remote PostgreSQL is configured with SSL support and a CA Root certificate file is not present, the certificate file must be copied into this directory on the host machine.</p> <p>Default value: <code>/opt/timeline/db-ssl</code></p> <p>PG_SSL_ROOT_CERT</p> <p>This is the path for a database CA Root certificate file.</p> <p>If your remote PostgreSQL is configured with SSL support and a CA Root certificate file used, a path to the root certificate file must be specified in this key.</p> <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 they are backed up regularly.</p> <p>Default value: <code>/opt/timeline/storage</code></p> <p>LICENSE</p> <p>The path to the directory where the license file is located relative to the installation directory.</p> <p>Default value: <code>/opt/timeline/license</code></p> <p>By default, all directories are under the installation directory.</p>
	<p>Example</p>	<pre>LOGS=/opt/timeline/logs NGINX_CONF=/opt/timeline/nginx STORAGE_DIR=/opt/timeline/storage LICENSE=/opt/timeline/license</pre>

HTTPS configuration

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 Process Intelligence. You can choose one of the following options:

- Use an 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.
- 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.

 Self-signed certificates are not recommended in production environments.

If you install the program in a Production environment, it is strongly recommended to use HTTPS and not HTTP.

To configure HTTPS:

1. Obtain an SSL certificate and a private key.
2. Run the Process Intelligence installation and follow the steps in the installation wizard. For more information, see [Install Process Intelligence](#).
 - a. To enable SSL between instances of the PostgreSQL database and application provide path to your database SSL certificate at the Database Connection step. If your PostgreSQL is configured using SSL, provide path to your SSL CA root certificate.
 - b. To enable SSL between the application and client, specify the HTTPS port and Base URL for the HTTPS port at the Web Server step.

3. After the Process Intelligence installation process is complete, do the following:
 - a. Find the **ssl.conf.tpl** and **ssl.conf** files in the `$TIMELINE_INSTALLATION_DIR/nginx` folder and rename the **ssl.conf.tpl** file to **ssl.conf**.
 - b. Copy your SSL certificate and private key files to the `$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**. 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`
 - iii. `DB_SSL`

If your remote PostgreSQL is configured with SSL support without root certificate make sure you certificate is in the specified folder.
 - iv. `DB_SSL=./db-ssl`

If your remote PostgreSQL is configured with SSL support and a CA Root certificate file used, make sure a path to the root certificate file is specified in this variable.
4. Restart the Process Intelligence application to apply all the changes:

```
systemctl restart timeline
```
5. [Perform a health check](#).

Network connection settings

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

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

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

Background upload

The background upload feature involves monitoring a folder for files copied into it. Whenever a new ZIP file is detected in that folder, the application grabs it and interprets it as an uploaded archive. You can configure the location of this folder in the [STORAGE variable in the .env file](#).

The file can be copied to the specified folder via 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 Linux systems, the `logrotate` command is usually already present by default (generally 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. Two typical use cases are described below:

When the log files are placed inside `/opt/timeline` or a similar location, 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  
}
```

2. 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.
3. This assumes that the Linux system already has a logrotation 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.

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
```

 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.

Patch installation

If the upload from an ODBC data source has failed and you are upgrading from Process Intelligence 1.1, you need to [download](#) and install the `Dockerfile.odbcpatch` file.

Prerequisites

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

Step 1 – Setup

1. HOST: Install the Process Intelligence application.
2. HOST: Stop the Process Intelligence service using the command `systemctl stop timeline`

Step 2 – Patch

Patch with internet connection on HOST.

HOST: Patch the timeline/backend image with the received dockerfile:

```
cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t  
timeline/backend:latest
```

Patch without internet connection on HOST.

1. HOST: Export the timeline/backend image:

```
docker save -o timeline-backend-latest.tar timeline/backend:latest
```

2. Move the exported timeline-backend-latest.tar to the BUILDER machine.

3. BUILDER: Load the timeline/backend image:

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

4. BUILDER: Patch the timeline/backend image with the received dockerfile:

```
cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t  
timeline/backend:latest
```

5. BUILDER: Export the patched timeline/backend image:

```
docker save -o timeline-backend-latest-patched.tar timeline/backend:latest
```

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

7. 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 the Process Intelligence service

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

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

```
docker image ls
```

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

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

3. HOST: Restart Process Intelligence service:

```
systemctl start timeline
```


Troubleshooting

Not enough space for Process Intelligence installation

When installing Process Intelligence, large docker images are loaded. If there is not enough space to load them, the installation might fail. This can also be caused by insufficient available RAM.

You are recommended to confirm that there is sufficient memory and space before proceeding with the installation. To do this:

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`

 If you have multiple partitions (such as, /var and /tmp in the example image below), those will also need free space, in addition to the target partition where you are planning to install Process Intelligence. It is recommended to have about 10 GB free on these partitions to ensure successful installation.

4. Check the [Process Intelligence system requirements](#) to confirm that you have enough memory and space for the installation.

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]#
```

Process Intelligence is not accessible outside the installed machine when using Red Hat Enterprise Linux

This issue only occurs on host machines running Red Hat Enterprise Linux, as its installation uses a 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`


If this occurs, recreate the Docker network as follows:

1. Stop the Process Intelligence service to make sure it does not try to use Docker using `service timeline stop`
2. Remove all containers using the command `docker container prune`. When asked for confirmation, select **Yes**.
3. List all the Docker networks using `docker network ls`

4. Inspect the Docker network to get its gateway IP using `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    },
```

5. Delete the network related to Process Intelligence using `docker network rm timeline_network`
6. Verify that the `timeline_network` has been permanently deleted using `docker network ls`
7. Stop the Docker service using `service docker stop`
8. Stop the PostgreSQL service using `service postgresql-12 stop`

 Perform this step only if you are using a local database.

9. Temporarily delete all firewall rules using `iptables --flush`
10. Restart the firewall using `service firewalld stop` and `service firewalld start`
11. Start the Docker service using `service docker start`
12. Create a new Docker network using `docker network create timeline_network`
13. Flush the IP tables again after restart using `iptables --flush`
14. Inspect the network using Docker. The IP address of the gateway might change: `network inspect timeline_network`
15. 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
```

16. Perform the following steps if you are using a local 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, for example, `host all all 172.10.0.1/24 md5`

The default path for PostgreSQL 12 conf files is `/var/lib/pgsql/12/data/`
 - ii. postgresql.conf

Update the listen address with the value of the new DOCKER_GATEWAY_IP.

The default path for PostgreSQL 12 conf files: `/var/lib/pgsql/12/data/`
 - b. Restart the PostgreSQL service using `service postgresql-12 start`

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`
17. Make sure that the Docker service is running using `service docker status`
18. Start the Process Intelligence service using `service timeline start`
19. Verify that all the containers are up and running (this can take some time) using `docker container ls -a`
20. Check that the Process Intelligence application is available. The following command should return an HTML response: `curl localhost`
21. Verify that the Process Intelligence application is available outside of the installed machine, by opening the application on the DNS/hostname of the machine.