
# AUTOMATIONEDGE SERVER

## INSTALLATION GUIDE

Release 7.7.4

https://automationedge.com

General: info@automationedge.com
Support: support@automationedge.com
Training: training.support@automationedge.com

India: +91-20-6632-1000
U.S.A. : +1-281-220-1057

# Contents

# About the Installation guide

The guide provides details about the installation process of the AutomationEdge Server (AE Server).

The chapter includes the following topics:



## Intended audience

The guide is useful for users who want to install the AE Server.

## Document conventions

| Conventions | Meaning | Example |
|---|---|---|
| **Bold** | The **bold** typeface is used to present references to menu options, fields, numbered captions, sections, and button names. | In the **Edit System Variable** dialog, enter the details. |
| *Italic* | The italic typeface is used to present keyboard or user entries. | Enter a new name for the system variable. For example, *Path*. |
| Bulleted list | A bulleted list is used for an unordered series of concepts, items, or options. | This chapter includes the following topics:<br>• Intended audience<br>• Document conventions |
| Numbered list | A numbered list indicates the sequence of processes, events, or steps. | 1. Under **System Variables**, select *Path* as the system variable, and click **Edit**.<br>2. Scroll to the beginning of the variable value and enter *%JRE_HOME%bin*. |
|  | The symbol indicates additional information. |  You can install the database and Apache ActiveMQ on different machines, if required. |

| Conventions | Meaning | Example |
|---|---|---|
|  | The symbol indicates very important information. |  If you decide to enable the **Workflow Monitoring** feature and deploy the **Workflow Monitoring** microservice, then Nginx will be the entry point for AutomationEdge. |
|  | The symbol indicates AutomationEdge's recommendation for users that are helpful in installation or application usage. |  Install Nginx and Metrics Service on separate machine than the one for Tomcat. |

# Prerequisites

In the chapter, you will learn about the prerequisites required for installing the AE Server.

## System prerequisites

In this topic, you will learn about the minimum system configurations required to install AE.

### Supported operating systems

Following is the list of requirements to deploy AutomationEdge (AE) server:

| | | | |
|---|---|---|---|
| **Operating system** | Windows Server 2012, 2016, and 2019 | Windows 10 (64 bit) | Linux (64 bit) |
| **Random Access Memory (RAM)** | 8 GB | | |
| **Hard Disk** | 200 GB | | |
| **No. of virtual CPUs** | 4 | | |

Following is the list of requirements for AE Agent deployment:

| | | | |
|---|---|---|---|
| **Operating system** | Windows 10 (64 bit), Windows 11 (64 bit) | Linux 64 bit | Windows server |
| **Random Access Memory (RAM)** | 4 GB | | |
| **Hard Disk** | 200 GB | | |
| **No. of virtual CPUs** | 2 | | |

### Required software

Following is the list of required software:

| Tool | Version | URL |
|---|---|---|
| Java Runtime Environment (JRE 64 bit) | For AE server release 6.0.0 onwards, Adopt JDK's HotSpot version 11.0.9 | https://adoptopenjdk.net/releases.html |
| Database | PostgreSQL 11, 12, 15 | https://www.enterprisedb.com/downloads/postgres-postgresql-downloads |
| | Oracle 19c | |
| | Microsoft SQL Server 2016, 2017, and 2019 | |
| Apache Tomcat for 64 bit | Tomcat 9 version 9.0.35 or later | https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.35/bin/ |
| Apache ActiveMQ | 5.18.0 | https://activemq.apache.org/activemq-5018000-release |

In the *Production Environment*, it is recommended to install the database and ActiveMQ on different machines, that is, machines other than where Tomcat is installed.

# Supported browsers

Following browsers are supported:

| Release version | Chrome | Microsoft Edge | Firefox |
|---|---|---|---|
| 7.7.4 | 57.0 and above | 92 and above | 52.0 and above |

# Pre-installation activities

In the chapter, you will learn about activities you need to perform before proceeding with the AE Server installation, and includes the following topic, Setup AE home directory structure.

## Setup AE home directory structure

Create an AE Home directory structure before you start with the AE Server installation.

### AE directory structure on Windows

Create various directories on Windows, as required.

#### To create AE directory structure on Windows:

1. Create the top-level directory of AE installation, which is also known as AE base directory on your system. For example, *AutomationEdge Base* → *<preferred_drive>:/AutomationEdge*.

   All AE software and components are installed in the AE base directory.

2. Create the following two directories after creating the AE base directory:

   *AutomationEdge Tools* → *<preferred_drive>:/AutomationEdge/tools*

   *AutomationEdge Home* → *<preferred_drive>:/AutomationEdge/aehome*

   - The complete file path of AE Home is referred to as <AE Home>. AE Home is a working directory for the AE Server.
   - Similarly, set up the directory structure in Linux environment.

### Setup AE home environment variable

After creating the AE Home directory, set its environment variable.

#### To create environment variable for Windows:

1. Access the **System Properties** dialog, click **Environment Variables…**, and the **Environment Variables** dialog appears.
2. Click **Edit…**, the **Edit System Variable** dialog appears. Enter the following details in the dialog:
   - **Variable name**: Enter as, *AE_Home*.
   - **Variable value**: Enter value as, *E:\AutomationEdge\aehome*.

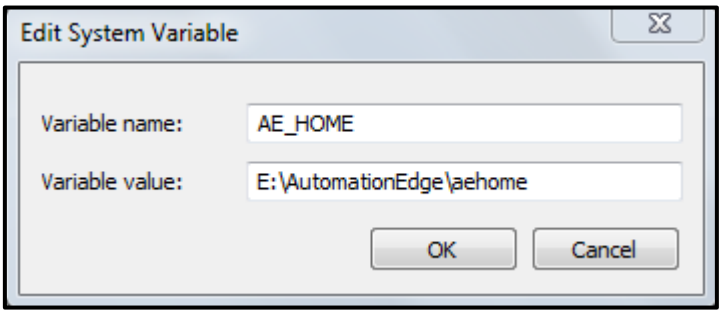



*Figure 1: Edit System Variable dialog*

To create environment variable for Linux, enter the following details: *AE_HOME=<path for the AE working directory>* in */etc/environment file or in /home/<user>/.bash_profile/.bash_rc/.profile*.

# Install AE Server

In the chapter, you will learn about the steps to install Java, database, and ActiveMQ required to install the AE Server.

The chapter includes the following topics:

- Install Java
- Install database
- Install Apache ActiveMQ

# Install Java

You need to install Java to work with AE Server.

## Setup Adopt JDK Hotspot

You need to setup Adopt JDK Hotspot version 11.0.9.

### To setup adopt JDK hotspot:

1. Access the binaries for the JDK at https://adoptopenjdk.net/releases.html.
2. Identify the JRE binaries for the operating system, such as Windows and Linux with desired architecture (x64 or x86).
3. Unzip the JRE binaries into a folder, **JRE_HOME for JRE 11**.
4. Add **JRE_HOME/bin** folder to the PATH environment variable at the start. It must be the first entry in the PATH.

## Setup JRE_HOME

Set the JRE_HOME environment variable to point to the JRE installation directory.

### To set environment variable:

1. Access the **System Properties** dialog, click **Environment Variables…**, and the **Environment Variables** dialog appears.
2. Click **Edit…**, the **Edit System Variable** dialog appears. Enter the following details in the dialog:
   - **Variable name**: Enter as, *JRE_HOME*.
   - **Variable value**: Enter value as, *New*.

- From AE release 6.0.0 onwards, point JRE_HOME to the JRE11 path.
- If you installed AE using the installer you must ensure that the AE_JRE_HOME environment variable points to the JRE installation directory.

## Edit Path environment variable

After creating the JRE_HOME variable, edit the **Path** environment variable.

### To edit Path environment variable:

1. In the **Environment Variables** dialog, select **Path** in the **System variables** section and click **Edit**.
2. In the **Edit environment variable** dialog, and enter *%JRE_HOME%bin* at the start of the path.

If you installed AE using installer your path variable must point to *%AE_JRE_HOME%/bin*.

## Verify installed Java

Verify the installed Java, using the following command:

```
java -version
```

For **Adopt Open JDK 11**, you will view the following message:

```
openjdk version "11.0.9" 2020-10-20
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.9+11)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.9+11, mixed mode)
```

# Install database

Install any or all of the following databases, as required:

- PostgreSQL
- Oracle
- Microsoft SQL Server

## Install PostgreSQL database

Install PostgreSQL database, which you have downloaded from https://www.enterprisedb.com/downloads/postgres-postgresql-downloads.

For PostgreSQL installation and administration details, see https://www.postgresql.org/docs/.

### To create a database:

1. After installing PostgreSQL, create a blank database, *vae*.
2. Create a user and role with granular access permissions, *read-write*.

Use the master user created when installing PostgreSQL for creating other users, roles, and databases only.

3. After creating the AE user, update **database.properties** file with the user credentials.

## Approach for setting up fine-grained access control in PostgreSQL

Following is the recommended approach for setting up fine-grained access control in PostgreSQL:

- Use the master user to create roles per application, such as read-write and read-only roles.
- Add permissions to allow these roles to access various database objects. For example, the read-only role can only run SELECT queries.
- Grant the roles the least possible permissions required for the functionality.
- Create new users for application or specific functionality. For example, *ae_app_user* and *reporting_user*.
- Assign the applicable roles to these users to quickly grant them the same permissions as the role. For example, grant the read-write role to *ae_app_user* and grant the read-only role to *reporting_user*.
- Revoke user permissions by removing the user's role.

# Install Oracle

Install Oracle database, if required.

### To install Oracle database:

1. Install Oracle database.

For Oracle installation and administration details, see https://docs.oracle.com/en/database/oracle/oracle-database/index.html.

2. After installing Oracle database or use an already installed database.
3. Create a user, *vae* and grant appropriate permission. For example, *read-write*.

# Install Microsoft SQL Server

Install Microsoft SQL Server, if required.

### To install Microsoft SQL Server:

1. Download Microsoft SQL Server, and apply the required service packs to install the MSSQL Server.

   For service pack details and download, see https://support.microsoft.com/en-in/help/3011465/fix-sequence-object-generates-duplicate-sequence-values-when-sql-server.

2. Create a database, *vae* on MSSQL Server.

# Install Apache ActiveMQ

Install or upgrade Apache ActiveMQ.

- If you already installed Apache ActiveMQ, upgrade it to version 5.18.0.
- For Apache ActiveMQ installation and administration details, see https://activemq.apache.org/getting-started.
- In case you are unable to start ActiveMQ 5.18.0 as a service you can try any of the following workarounds:
  - (Recommended) Uninstall the existing ActiveMQ service and reinstall in a path which does not contain white spaces, preferably other than C drive.

  OR

  - Edit ACTIVEMQ_HOME\bin\win64\wrapper.conf and modify the property value for

```
wrapper.java.additional.13=-
Djolokia.conf=file:"%ACTIVEMQ_CONF%/jolokia-access.xml"
```

# Install Apache Tomcat

Install Apache Tomcat and configure the details.

### To install Apache Tomcat:

1. Download Apache Tomcat for 64-bit from https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.35/bin/.
2. Install Apache Tomcat and configure the details, as required.

For Apache Tomcat installation and administration details, see https://tomcat.apache.org/tomcat-9.0-doc/index.html.

After Apache Tomcat installation and configurations, complete the following configurations:

## Delete Tomcat OOTB applications

Before deploying AE apps, user must delete the out-of-the-box apps, such as ROOT, manager, docs, and so on that Tomcat installation packages provide.

### To delete the apps:

1. Go to **<Tomcat home>/webapps**.
2. Delete following directories: docs, examples, host-manager, manager, and ROOT.

## Make aeui the default application

Configure Tomcat's default settings to make aeui the default application.

### To configure the default application settings:

1. Add docBase detail in the **<Tomcat home>/conf/server.xml** file. In the **<Host></Host>** tags, enter:

```
<Context path="" docBase="aeui"/>
```

2. Edit **<TOMCAT_HOME>/conf/web.xml** file to redirect to the aeui application. Before the **</web-app>** tag, enter:

```
<!-- Require HTTPS for everything except /img (favicon) and css -->
<security-constraint>
<web-resource-collection>
<web-resource-name>HTTPSOnly</web-resource-name>
<url-pattern>/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
<security-constraint>
<web-resource-collection>
<web-resource-name>HTTPSOrHTTP</web-resource-name>
<url-pattern>*.ico</url-pattern>
<url-pattern>/img/*</url-pattern>
<url-pattern>/css/*</url-pattern>
```

```
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>NONE</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

3. After all the steps are complete, start or restart Tomcat server, and check if the application is running on SSL. It must also redirect the http request to https automatically.

## Hide server name and version

Hide the Tomcat server name and version. Complete the process of hiding the details and then restart the Tomcat server.

### To hide Tomcat server name and version:

1. Go to **<Tomcat home>/lib**, and create the directory **org/apache/catalina/util** under **lib**.
2. In the new directory, create a **ServerInfo.properties** file, and add an entry for **server.info** property in the file.

> Keep the value for this property blank. For example, **server.info**=.

3. Edit **server.xml** configuration file available under **<Tomcat home>/conf**, and add the following details in the **Connector** node:

> The step is optional.

*server="AutomationEdge"*

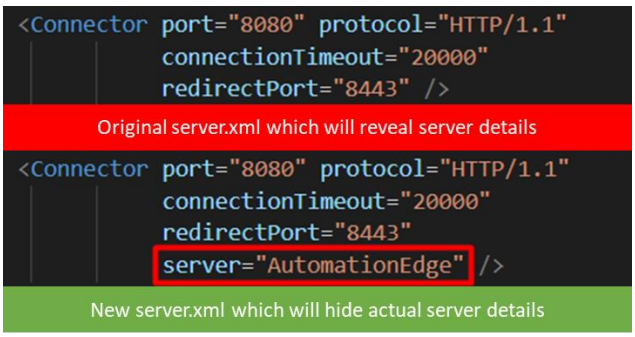



*Figure 2: Connector node view*

# Tomcat memory settings

You can set Tomcat memory settings in the following two ways:

- Memory settings for Tomcat as service
- Memory settings for Tomcat using command line

The memory setting for Tomcat is mandatory.

## Memory settings for Tomcat as service

When running Tomcat as service, modify the memory settings, as required.

### To configure Tomcat as service details:

1. Stop Tomcat service, if it is up.
2. Go to **%TOMCAT_HOME%/bin**, and select the Tomcat executable, and run it. For example, **tomcat9w.exe**.
3. Under the Java tab, change the following properties:
   - **Initial memory pool**: *1024 MB**
   - **Maximum memory pool**: *2048 MB**

You can increase the allocated memory to higher values depending upon the load on the AE server. For example, if a machine has 8 GB RAM set, the Initial memory pool: *2048 MB** and Maximum memory pool: *4096 MB**.

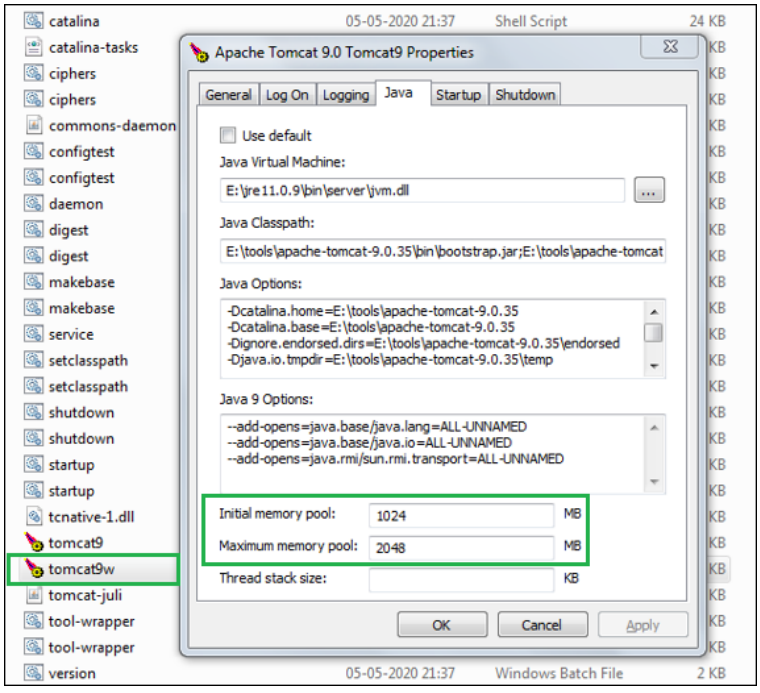
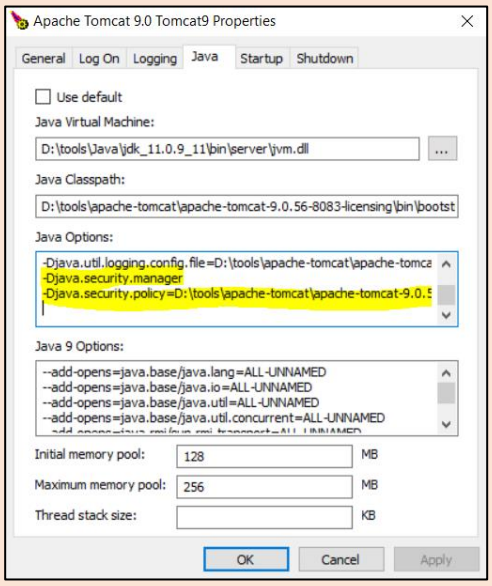



*Figure 3: Apache Tomcat 9.0 Tomcat9 Properties dialog*

4. Start or restart the Tomcat Service.

- For details about starting the Tomcat Security Manager, see Enable Apache Tomcat Security Manager.
- Ensure you use the absolute path to the policy file. For example, see the following screenshot:




## Memory settings for Tomcat using command line

When running Tomcat using command line, modify the memory settings, as required.

### To configure Tomcat memory settings:

1. Stop Tomcat service, if it is up.
2. Create **setenv.bat** or **setenv.sh** for Windows and Linux respectively in **%TOMCAT_HOME%/bin**.
3. Add the following detail to the **setenv** file:
   - For Windows set *JAVA_OPTS=-Xms1024m -Xmx2048m*.
   - For Linux export *JAVA_OPTS="-Xms1024m -Xmx2048m"*.

   You can increase the allocated memory to higher values depending upon the load on the AE server.
4. Start or restart Tomcat.

# Verify and start services

Change the **Startup Type** to *Automatic* for the following Windows services and start them in the following order:

1. Database installed for AE, such as PostgreSQL
2. ActiveMQ
3. ApacheTomcat, the version installed for AE

# Configure AutomationEdge environment

In the chapter, you will learn about configuring the AutomationEdge environment.

The chapter includes the following topics:

- Deploy .war files
- Configure application settings
- Configure AutomationEdge UI settings
- Verify AutomationEdge UI sign-in

# Deploy .war files

Deploy .war files for configuring AE environment.

### To deploy .war files:

1. Stop Tomcat server.
2. Copy the following .war files in **<Tomcat home>/webapps**.
   - aeengine.war
   - aeui.war
3. Restart Tomcat service to extract .war files in **<Tomcat home>/webapps** directory.

# Configure application settings

After deploying the applications, configure their settings.

## AutomationEdge Engine Settings

After deployment of applications, a directory named **/conf** is created under <AE home>. It has two files, **ae.properties** and **database.properties**. Configure their properties with appropriate values.

### Configure ae.properties

Set the following properties with appropriate values:

**Part 1: ActiveMQ Broker URL**

1. Check value for ActiveMQ Server IP or host, and modify it only if required. For example, *activemq.broker.url=tcp://localhost:61616*.

2. If authentication is enabled for ActiveMQ, then provide **mq.username** and **mq.password**, else keep it blank.

Plain text password will be encrypted by AE.

**Part 2: Session Timeout**

You can configure the session timeout settings in the AE application.

The session timeout setting is optional.

To modify the session timeout value, go to **ae.sessiontoken.validityInMinutes** property, and change the default value.

For example, **ae.sessiontoken.validityInMinutes**=*20*

The default session timeout value is 15 minutes.

**Part 3: Session Token Validity**

You can set the session token validity in hours. For example, if the session token validity value is set to 5 then the session token expires after 5 hours irrespective of continuous user activities.

To modify the session token validity value, go to **ae.sessiontoken.autoRenewalInHours** property, and change the default value. The default value of **ae.sessiontoken.autoRenewalInHours** = *-1*, which means there is no expiry.

**Part 4: Supported file extensions for upload**

**ae.banned.file.extension** contains the comma separated list of file extensions to be restricted from upload, such as *.msi*, *.exe*. In addition to the default extensions, you can provide a list of extensions you want to restrict from uploading.

The **ae.validate.files.in.archive** property allows you to set restrictions on files with specific extensions from uploading to the AE server. Set the flag to *true*. The default flag is *false*.

```
ae.validate.files.in.archive = true
```

If a file with the banned extension is in an archive, for example, **Archive.zip/help.html**, then an exception is thrown when uploading to the AE server.

However, if a file with the banned extension is in a nested archive, then the file gets uploaded to the AE server. For example,

**Archive.zip/help.rar/help.html**: The settings are ignored and the file is uploaded to the AE server.

**Part 5: Pending Requests Threshold**

For the **Pending Requests Threshold** feature to be available, configure the details under **Settings** → **Email Notification**, and specify the users for sending email notification. You can configure the lower and higher threshold values for requests.

If the agent is down, requests go in New State for lower and higher threshold. However, if number of pending requests goes over the lower threshold, then the users receive an email notification.

For example, the lower threshold value for **ae.pendingreq.lower.threshold** = *300.*

Users can submit requests even if the request count crosses the lower threshold value.

For example, the higher threshold value for **ae.pendingreq.higher.threshold** = *800.*

However, if number of pending requests goes over the higher threshold, then more requests can be submitted.

**Part 6: Pending request limit per tenant**

The request limit feature is enabled if the property value is greater than *0*. If its value is less than or equal to 0 then a tenant level request limit is imposed. If the tenant level limit, new + retry, is crossed then a notification is sent to Tenant Admins and System Admins.

If the request is submitted by scheduler, then a record is created with *Failure* status and proper error message is displayed to the users.

```
ae.pending.request.limit.per.tenant = 0
```

**Part 7: Archive Location**

- **ae.archive.location**

By default, the archived zipped csv files are available in default location, **AE_Home\Archives**.

You can configure the default purging destination of the archived files. Set the parameters in the **ae.properties** file for the **ae.archive.location** property. For example, **ae.archive.location** = *<Desired filepath(e.g D:/testPurge/Archives)>*.

- Tenant folders are created in the purging destination.
- The zip files include purging data of all the four entities, that is, workflow requests, audit logs notification history, and user session history.

In the file path, use the forward (/) slash, and if you are using the back (\) slash, use two slashes (\\).

**Part 8: Sysadmin forget password feature**

Enable the sysadmin forget password feature by setting this to true.

**ae.forgot.password.enabled**=*true*

**Part 9: High Availability (HA) properties**

Following are the default values of the properties:

**ae.clusters.members**=*localhost*

**ae.clusters.port**=*5900*

In case of HA, set following properties:

Value of **ae.clusters.members** must be comma separated list of host and IP addresses of nodes in the HA cluster. For example, if there are 3 nodes in HA then property values can be:

**ae.clusters.members**=*10.41.4.1*, *10.41.4.2*, *10.41.4.3*

**ae.clusters.port**=*5900*

Do not modify the default properties if there is no HA required.

**Part 10: Cleanup Requests in New state**

Cleanup job frequency in minutes for requests in *New* state.

Default value is 30 minutes. Min value can be 15 minutes and Max value can be 60 minutes.

```
ae.new.request.cleanup.job.interval.minutes=30
```

**Part 11: Response limit for JMS messages**

The property is related to Integration Service. It is to specify number of messages to dequeue at a time from JMS Completed Queue.

```
ae.jms.completed.queue.receive.limit= 10
```

**Part 12: JMS expire message in days**

It is the property to specify expiry of messages from JMS Completed Queue. Messages will be deleted from JMS queue, so that queue does not grow infinitely. Default value is 7 days.

```
ae.jms.completed.queue.expire.period.days=7
```

**Part 13: Purging Archive**

Specify the complete file path of purged records.

```
ae.archive.location=
```

**Part 14: Scheduler**

The **ae.scheduler.threadpool.size** property is set while initializing the scheduler. The property indicates in an AE Scheduler Thread Pool size, that is, thread count in the Thread Pool. The default value can be increased to accommodate higher number of Scheduled Requests at one instance of time.

```
ae.scheduler.threadpool.size = 20 (Allowed range 10-100)
```

**Part 15: SMTP server timeout settings**

The **ae.mail.smtp.timeout.seconds** property specifies Java Mail SMTP timeout value in seconds. The property indicates the time in which a task will be terminated if a response is not received from the SMTP server. Default timeout value is *30* seconds.

```
ae.mail.smtp.timeout.seconds = 30
```

The **ae.mail.smtp.connectiontimeout.seconds** property specifies Java Mail SMTP connection timeout value in seconds. The property indicates the time in which the SMTP mail connection will be terminated if it remains idle for more than the specified time. Default connection timeout value is *30* seconds.

```
ae.mail.smtp.connectiontimeout.seconds = 30
```

**Part 16: Scan embedded files settings**

The **ae.scan.embedded.files** property specifies if the file being uploaded needs to be scanned for embedded files. Default setting is *false*, which indicates that the file will not be scanned for embedded files.

```
ae.scan.embedded.files  =  false
```

**Part 17: Workflow monitoring**

The **ae.workflow.monitoring.enabled** property enables workflow monitoring through AE UI. Default setting is *false*. To enable the property, change the flag to *true*.

For details about deploying the workflow monitoring microservice, see **Workflow Monitoring Deployment guide**.

```
ae.workflow.monitoring.enabled = true
```

**Part 18: Schedules**

Add the **ae.schedules.enable.cron.expression.input** property, if required. If you add the property and set the flag as *true*, then you can provide **Cron expression** in **Scheduler → Create New Schedule** through AE UI.

```
ae.schedules.enable.cron.expression.input = true
```

## Configure database.properties

Set the following properties with appropriate values depending on the installed database.

1. **PostgreSQL**

```
database.type=POSTGRESQL
database.url=jdbc:postgresql://localhost:5432/vae?profileSQL=false

database.username=postgres

database.password=<Password for postgres user set during PostgreSQL
installation>
```

2. **Oracle**

```
database.type=ORACLE
database.url=jdbc:oracle:thin:@localhost:1521:orcl

database.username=vae

database.password=<Password for vae user set during Oracle user creation.>
```

3. **MSSQL**

```
database.type=MSSQLSERVER

database.url=jdbc:sqlserver://localhost:1433;databaseName=vae;

database.username=sa

database.password==<Password for vae user set during MSSQL User creation>
```

# Configure AutomationEdge UI settings

Configure the AE UI settings in the **<Tomcat home>/webapps/aeui/aeui-config.properties** file.

By default, after AE installation **aeui-config.properties** file is in plain text format.

## To change the UI settings:

1. Access **aeui-config.properties** file in the **<Tomcat home>/webapps/aeui** directory.
2. In the **aeui-config.properties** file configure the following properties:
   i. **AutomationEdge Base URL**

   If **aeengine** and **aeui** are deployed on two separate Tomcat servers on two different machines, then change the **baseurl** property with appropriate **IP** or **Host** and **Port** values where AE server (aeengine) is running.

   Default value for the **baseurl** property is, **baseurl** = */aeengine/rest*.
   For example, **baseurl**=*https://10.2.4.56:8080/aeengine/rest*.

   **ii. Customer Logo**

   To view the customer logo in the UI application, add customer logo file under directory **<Tomcat home>/webapps/aeui/assets/images**.

File types supported are, *.jpeg*, *.png*, *.svg*, and *.gif*.

Update the property **tenantLogoFile** with the file name you have added in the previous step.

Default value is, **tenantLogoFile** = *customer-logo.png*.

**iii. Login Message**

To add custom login message that appears on the login screen, enter the text message in the **loginMessage** property. For example, **loginMessage**=*Welcome to AutomationEdge*.

Default value for the property **loginMessage** is blank.

iv. **Storage**

Use the property to store browser session data.

For security reasons, the default and recommended option is *InMemory*, which means that session data is not stored on disk and is lost on browser refresh. The other available option is *LocalStorage*.

Default value for the **storage** property is **storage** = *InMemory*.

**v. Security vault**

Use the property to enable the System Administrator to assign vaults to tenants.

Default value for the **security vault** property is **enableSecurityVault** = *false*.

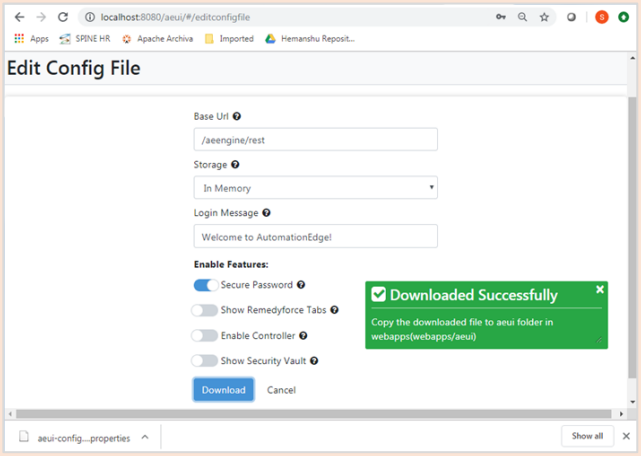
vi. **Captcha**

Use the property to enable captcha text on the AE login page. *Enabling the feature is optional.*

Default value for the **captcha** property is **enableCaptcha** = *false*.

To store **aeui-config.properties** in encrypted format:

1. Access the following URL, http://<AE server host>:8080/aeui/#/editconfigfile**.**
2. Configure the required AEUI properties and download encrypted file.




3. Replace the plain text file with the encrypted file.

# Verify AutomationEdge UI sign-in

Configure all properties and verify the details as required by the environment, and then restart Tomcat service.

## To verify AE UI sign-in:

1. Open any web browser. For example, Google Chrome.
2. Enter URL as *<http or https>://<IP or Host>:<PORT >/aeui*.
3. Sign in with the following user credentials, configured as part of seed data:

   **username**: *sysadmin*

   **password**: *vY78_h$ia*

If the sign-in is successful, reset the password for the Sysadmin user. After successful password reset, sign in again with the new password.

# Post installation activities

In the chapter, you will learn about post installation activities that you need to perform to use the AE Server.

The chapter includes the following topic: AE server post installation

## AE server post installation

After installing the AE Server, sign-in as the System Administrator and complete the following AE post installation tasks:

### To complete post installation activities:

1. Access the AE Server, *<http or https>://<IP or Host>:<PORT >/aeui*.
2. Sign in with your System Administrator credentials, and then set the security questions for enabling forgot password option.
3. Update **System Settings** by setting values for **Server URL**, **Cleanup Requests**, and so on.
4. Upload Process Studio zip as an artefact.
5. Create a new Tenant and a new user with *Admin* role.
6. Upload the plugins and assign to Tenants.
7. Update the Sysadmin policy as required.

For more information on AE Server post installation steps, see **AutomationEdge System Administrator's guide** and **AutomationEdge User's guide**.

# Security Hardening

Security Hardening is the process of enhancing a server's or application's security by following the best practices and standards, which results in a very secure operating environment.

For more information, see https://docs.bmc.com/docs/security/basic-tomcat-security-configuration-recommendations-924057229.html.

## Configure Tomcat with TLS (HTTPS)

For configuring Tomcat with TLS(HTTPS), see the Apache Tomcat documentation at https://tomcat.apache.org/tomcat-9.0-doc/index.html.

Following are links to some useful topics in the documentation:

- https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html
- https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html

## Security settings for ActiveMQ

For configuring Apache ActiveMQ security settings, see the following links:

- https://activemq.apache.org/how-do-i-use-ssl
- https://activemq.apache.org/ssl-transport-reference
- https://activemq.apache.org/security
- https://activemq.apache.org/encrypted-passwords.html
- https://activemq.apache.org/using-activemq

After completion of security hardening activities configure the ActiveMQ authentication setup for AE.

### ActiveMQ authentication for AE

ActiveMQ authentication is not enabled by default for AE. Configure the following settings in the AE engine.

- Simple authentication plugin is used to set authentication for ActiveMQ. Encrypt the password and store it safely in configuration files. Following is the encrypt password command:

  *$bin/activemq encrypt –password activemq –input mypassword*

- Where the password you want to encrypt is passed with the input argument, the password arguments is a secret used by the encryption.

- You get an encrypted password in the output. For example, **Encrypted password:** *K6lz7RKJg9DzGiX8eRbvUw==*
- In the **<ActiveMQ home>/conf/credentials-enc.properties** file, add the following: *aeuser.password=ENC(K6lz7RKJg9DzGiX8eRbvUw==)*
- You need to edit the **<ActiveMQ home>/conf/activemq.xml** file.
  - Delete the following bean definition:

```
<bean
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"
>
<property name="locations">
         <value>file:${activemq.conf}/credentials.properties</value>
</property>
</bean>
```

  - Add the following bean definitions:

```
<bean id="environmentVariablesConfiguration"
class="org.jasypt.encryption.pbe.config.EnvironmentStringPBEConfig">
         <property name="algorithm" value="PBEWithMD5AndDES" />
<property name="passwordEnvName" value="ACTIVEMQ_ENCRYPTION_PASSWORD"/>
</bean>
<bean id="configurationEncryptor"
class="org.jasypt.encryption.pbe.StandardPBEStringEncryptor">
         <property name="config" ref="environmentVariablesConfiguration" />
</bean>
<bean id="propertyConfigurer"
class="org.jasypt.spring31.properties.EncryptablePropertyPlaceholderConfigurer
">
<constructor-arg ref="configurationEncryptor" />
<property name="location" value="file:${activemq.conf}/credentials-
enc.properties"/>
</bean>
```

  - Add **<simpleAuthenticationPlugin>** tag under broker element in the **activemq.xml** file.

```
<bean id="environmentVariablesConfiguration"
class="org.jasypt.encryption.pbe.config.EnvironmentStringPBEConfig">
         <property name="algorithm" value="PBEWithMD5AndDES" />
<property name="passwordEnvName" value="ACTIVEMQ_ENCRYPTION_PASSWORD"/>
</bean>
<bean id="configurationEncryptor"
class="org.jasypt.encryption.pbe.StandardPBEStringEncryptor">
```

```
        <property name="config" ref="environmentVariablesConfiguration" />
</bean>
<bean id="propertyConfigurer"
class="org.jasypt.spring31.properties.EncryptablePropertyPlaceholderConfigurer
">
<constructor-arg ref="configurationEncryptor" />
<property name="location" value="file:${activemq.conf}/credentials-
enc.properties"/>
</bean>
```

- Set up ACTIVEMQ_ENCRYPTION_PASSWORD.

  After you have configured the authentication for ActiveMQ, set the `ACTIVEMQ_ENCRYPTION_PASSWORD` environment variable.

  - To set environment variable on **Windows** machine:

    a. Under **System Variables**, add **New** variable.

    b. Enter the variable name as **ACTIVEMQ_ENCRYPTION_PASSWORD** and enter *activemq* as the variable value. For example, **ACTIVEMQ_ENCRYPTION_PASSWORD**=*activemq*.

  - To set environment variable on **Linux** machine:

    Ensure you have the root privilege to create a file under **/etc/profile.d**.

    i. Create new file under directory **/etc/profile.d** which will set the environment variable. Following is the syntax for creating a file:

    *$cd /etc/profile.d/*

    *$sudo touch automationedge.sh*

    Creating the new file enables all users using Linux machines to access the environment variable with ease.

    ii. Edit **automationedge.sh** and add the following:

    *export ACTIVEMQ_ENCRYPTION_PASSWORD=activemq*

    iii. Save the file and restart the machine.

- To edit **ae.properties** file in **<AE home>**.

  i. Open **ae.properties** file and find properties **mq.username** and **mq.password**.

  **mq.username** and **mq.password** properties are empty by default.

ii. Enter the following details under **ActiveMQ credentials**.

*#ActiveMQ credentials*

*mq.username=aeuser*

*mq.password=mypassword*

- Restart services for ActiveMQ and Tomcat.

# AE with SSL

AE supports secure URL. As AE is hosted on Apache Tomcat, modify the configuration to secure Tomcat server with TLS.

You don't need to modify the configuration, if you have already completed configuring Tomcat for TLS (HTTPS).

# Migrate from previous releases

In this section we will discuss migration of AutomationEdge from previous releases to the current release.

- If you are migrating from previous AE release to 7.5.*, then it is recommended to upgrade to ActiveMQ 5.18.0 to mitigate vulnerabilities present in earlier versions of ActiveMQ.
- While you can migrate directly to version 7.* from any releases prior to 5.6.*, we strongly recommend you to go with the following migration step for a simplified and easier migration process:
    - To migrate from releases prior to 5.6.* to version 7.0.0 and above, you can first migrate to 5.6.* and then again migrate to the latest version of AE. For example, if your current version is 4.0 and you want to migrate to the latest 7.* version, then first migrate to 5.6.*, and then again migrate to the latest 7.* version.

The chapter includes the following topics:



## Migration Scenarios

The following table is a quick reference to the migration scenarios.

| Migration Scenarios | Version |
|---|---|
| Migrating from Releases prior to 5.4 | • Stop Agents<br>• Setup JRE 11 for AEEngine/Tomcat<br>• Migrate AEEngine<br>• Setup JRE 11 on Agent machines<br>• Start Agents.<br>• In case of Agent Already Logged in Error - wait for Agent to go to unknown state and Start Agent again. |

| Migration Scenarios | Version |
|---|---|
| Migrating from Release 5.4 and above – with "Include JRE with Agent" enabled | • Enable "Include JRE with Agent" (Navigation path - Agents→Agent Settings on AE UI) for your Tenant<br>• Setup JRE 11 for AEEngine/Tomcat<br>• Migrate AEEngine<br>• Agents are automatically upgraded |
| Migrating from Release 5.4 and above - with "Include JRE with Agent" disabled | If you don't want to use "Include JRE with Agent" and is disabled, see - **Migrating from Releases prior to 5.4**. |

# Pre-Migration Activities

Following are the pre-migration steps and setups for migration to AutomationEdge Release 7.0.0.

## Artefacts

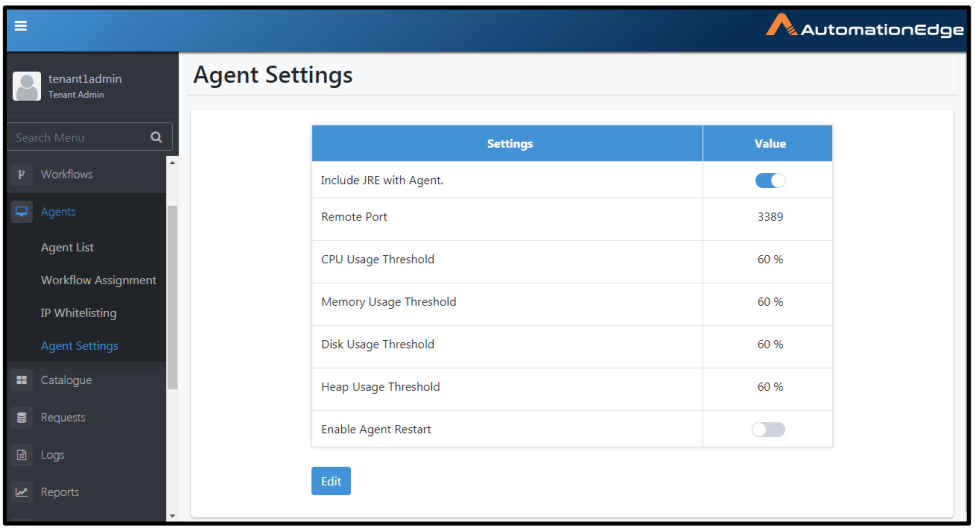
Make sure you have all the artefacts of the latest AE release and a valid AE license.

## Pre-migration activities for Agents

Following are the pre-migration activities for Agents,

1. AutomationEdge R6.0.0 onwards requires JRE 11. AutomationEdge Agents with AE 6.0.0 onwards also require JRE 11.
    - Agents can be bundled with JRE 11 from AutomationEdge server (preferred approach) only for R5.4.0 and above.
    - Agents can use system JRE 11.

    It is recommended to follow the first approach above. The bundled JRE in Agent is automatically upgraded from JRE binaries on AE server in future upgrades as well.
2. Following are the actions so that Agents are bundled with JRE 11 from AutomationEdge server for R5.4.0 and above.
    i. Enable the "Include JRE with Agent" (Navigation path - Agents→Agent Settings on AE UI) if not enabled already.




When the System Migrates to 7.0.0, Agents will automatically upgrade and also migrate to JRE11 supplied by AE.

ii. Stopping Agents:

- For AutomationEdge releases prior to 5.4.0, stop all Agents manually as a first step of migration.
- For Release 5.4.0 onwards Stopping Agents is optional - provided 'Include JRE with Agent' (option under Agents → Agent Settings) is enabled.

# Requests

Ensure that none of the Requests are in New or Execution Started Status.

# Backup AutomationEdge database

It is recommended to backup of the database. For example, database named 'vae' in several cases.

# Stop Tomcat Service

Before following the steps below, please stop Tomcat server.

# JRE

Setup the environment with OpenJDK 11 JRE (Adopt JDK's HotSpot version 11.0.9) for Tomcat/ActiveMQ for AutomationEdge R6.0.0 onwards. For details, see Install Java.

# Active MQ

- If ActiveMQ is using RDBMS storage, follow these steps as mentioned in ActiveMQ about adding 4 jars from https://mvnrepository.com/.

- Start ActiveMQ and test by storing sample message from the admin console.

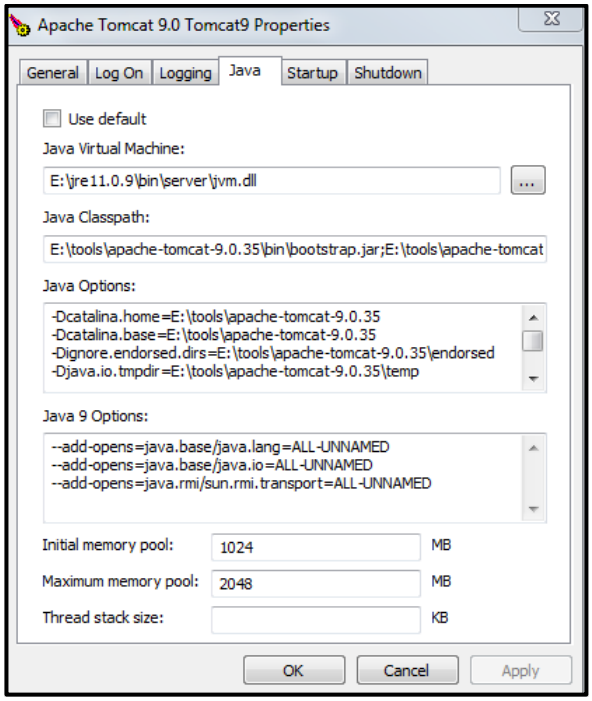
## Tomcat Services with Java 11:

If Tomcat services have not been configured for JRE 11 perform the following.

Tomcat Services need to be either recreated or reconfigured after migration to JRE 11 version.

Following are the steps to reconfigure Tomcat:

- Stop Tomcat if not already stopped.
- Locate and open execute Tomcat Monitor Application <TomcatServiceName>w.exe (e.g., tomcar9w.exe or AE-Tomcat-9.0.35w.exe in case AE is installed using AE installer etc.) available at <Tomcat Home\bin> (e.g., E:\tools\apache-tomcat-9.0.35\bin\).
- It opens the Tomcat Monitor Application.
- Following window appears, change the Java Virtual Machine Path to point to the JRE 11 jvm.dll (available at <JRE_HOME>/bin/server)




# AutomationEdge Server migration

## War Files Deployment

**Backups**

- It is recommended to take a backup of the following,

- Backup AutomationEdge database. You may refer AutomationEdge_R7.7.4_Postgre_SQL_Backup guide for Postgres database backup.
- Backup AutomationEdge home.
- Backup AutomationEdge internal configuration files from <Tomcat home>/webapps/aeengine/WEB-INF/classes, especially if they were modified to suit the customer environment.
    - application.properties
    - hibernate.properties
    - restvalidation.json
    - configuration.properties

**Delete files**

- *Delete vae.war/aeengine.war*, *vaeinterface.war/remedyforceinterface.war* , aeintegrationservice.war and *aeui.war* from <Tomcat home>/webapps directory.
- Delete *remedyforceinterface.war* (if present. Relevant only if *remedyforceinterface.war* is deployed) from <Tomcat home>/webapps directory.
- Delete directories vae/aeengine, vaeinterface/remedyforceinterface and aeui from <Tomcat home>/webapps

**Copy Files**

Once all the war files and the corresponding directories are deleted, copy the new war files ***aeengine.war*** and ***aeui.war*** to <Tomcat home>/webapps

**Start Tomcat**

Start the Tomcat Service so that the war files are extracted in **webapps** directory.

**Update Files**

Update the following internal properties files in **<Tomcat home>/webapps/aeengine/WEB-INF/classes**.

- application.properties
- hibernate.properties
- restvalidation.json
- configuration.properties

Do not replace these files from the backup. It may result in unstable application deployment.

**Restart Tomcat**

Restart the Tomcat service if there are any updates to the internal properties files above to make the changes effective.

**Security Hardening**

If settings are not done previously, for details, see Security Hardening.

# Post Migration Steps

In the section, we discuss the post migration steps required with respect to - License, Agents, Plugins & Workflows and Workflow Schedules.

## License

In case you do not have a compatible license, login with Tenant Administrator and apply AE license, compatible with the current release, that is, AE License 5.4.0 onwards; to use license features like Advanced Agents, Turbo Agents, and Assisted Agents.

## Agents

Agent post migration steps are required only if the steps Pre-migration activities for Agents are not performed.

In case of Agent upgradation failure, use any of the following two options. The recommended is **Option 1**.

### Option 1 (Agent is bundled with JRE from AE server - preferred option)

The option is for migrations from R5.4.0 and above.

Complete the following settings in AE UI, if **Include JRE with Agent** is not enabled:

- Enable “Include JRE with the Agent” (Navigation Settings →Agent Settings on AE UI) if not already done in the pre-migration Steps,
- If case Agents throw an error during upgradation, then set the following property-
    - Open the `application.properties` file present under `AGENT_HOME/conf folder`
    - Change the property `agent.upgrade.status` value to `Successful`

### Option 2 (Agent uses System JRE)

This option is required for releases prior to 5.4.

This option can also be used for releases 5.4 and above if desired but option 1 is preferable.

If "Include JRE with Agent" is disabled before and even after AE Server migration, then Agent can use System JRE.

Complete the following steps:

- Setup JRE 11 on the Agent machines manually. For details, see Install Java.
- Open the `application.properties` file present under `AGENT_HOME/conf folder.`
- Check `agent.upgrade.status.` If the value is other than Successful, change the value to ***Successful***

### Start Agents

- Any stopped Agents prior to upgrade should be started.
- You may start the Agents post migration of AutomationEdge server to 7.0.0, after completing settings mentioned in the sections 0/0 above - as required.
    - Agents will auto upgrade and JRE 11 will also be downloaded along with Agent binaries. The bundled JRE in Agent will be automatically upgraded from JRE binaries on AE server in future upgrades as well.
    - In case Agents are using System JRE, Agents are upgraded (apart from bundled JRE 11).

In case of releases 5.3 and prior, in case Agents were not stopped prior to upgrade, Agents will throw - Agent Already Logged in Error - wait for agent to go to unknown state and start Agent.

### Post Agent Startup

- All Agents start as Standard Agents. You may edit the Agents to Advanced or Turbo Agents. You may change Agent Mode depending on availability in license subscribed. However, Agent Mode cannot be changed when an Agent is in Unknown state.
- Agents acquire Tenant level Proxy settings if any during upgrades, discarding the Agent level proxy settings. The required changes need to be done manually.
- If you are running Agent as Administrator during restart after upgrade it does not start Agent as Administrator. If you wish to run the Agent as Administrator, once all the Agents are upgraded you need to shut the Agent and restart Agent as Administrator.

## Plugins & Workflows

- As a part of a migration to this Release 7.0.0, all the Process Studio workflows are made inactive.
- This is due to framework changes. If server is updated such that plugins do not match supporting framework versions on AE server, all the workflows using such plugins are disabled. User will be able to enable these workflows after new plugins are uploaded.

- User should upload the new zip of Plugins 2.0 or higher provided in the release package. Until the new plugins are uploaded onto the server, user will not be able to make the workflows active again. Once new plugins are uploaded the workflows can be enabled*.
- Login with System Administrator and upload Plugins zip files.

- Now activate all workflows.

When uploading PS-plugins using zip, you may get errors for **OOTB**, **Advanced Rest Client**, or **WebGUI** plugins, stating *Duplicate step or process entry name*.

**Workaround**:

Plugins distribution contains the following zip files:

- Data-plugins-3.0.zip
- IT-plugins-3.0.zip
- ITSM-plugins-3.0.zip
- MachineLearning-plugins-3.0.zip
- PS-plugins-3.0.zip
- Script-plugins-3.0.zip
- Cloud-Storage-3.0.zip

The PS-plugins-3.0.zip contains several jars, including:

- gui-automation-2.0.jar
- advanced-rest-client-3.0.jar

For PS-plugins.zip you might get errors while uploading. There are two possible workarounds as follows. You may follow any one of these.

- Upload two new plugin jars followed by PS-plugins.zip
  - Upload gui-automation plugin Jar (gui-automation-3.0.jar) independently
  - Then upload advanced rest client plugin jar (advanced-rest-client-3.0.jar)
  - Then use the PS-plugins-2.0.zip to update rest of the plugins
  - Upload PS-plugins-2.0.zip again and again, until all plugin specific errors are resolved. Typically, you might have to upload it thrice.

For upgrades from 4.0.1.4 to 7.0.0 - Any exported AutomationEdge workflows from the migrated 7.0.0., instance cannot be imported to other AE instances (e.g. Development to UAT and Production) if workflow was created in 4.0.1.4. Such a

migrated workflow needs to be opened in Process Studio and created in other instances with the Publish option.

## Schedules

Repeat interval values for minutes from Release 5.4.0 onwards, is restricted to 1, 2, 3, 4, 5, 10, 15, 20, 30, 45. As a part of AE Server migration, Schedules having minute's interval other than aforementioned allowed values are disabled. Users need to update the interval with valid values after migration and then enable such schedules.

## Database Drivers

We ship JDBC jars for all three databases we support i.e., PostgreSQL, Oracle and MSSQL Server with application.

- There is no need to put the database JDBC jar separately into TOMCAT_HOME/lib (in case of Oracle/ MSSQL Server).
- In case of migration of AutomationEdge from earlier versions 3.2.* and 4.0.*, stop services and remove JDBC jar of Oracle/MSSql Server from TOMCAT_HOME/lib if put earlier so as to avoid conflicts. Restart the services.

# APPENDICES

# Appendix 1: AE file system

The appendix lists some important files in AE file system.

Following are the components of the AE Tools directory:

1. **Active MQ HOME**
   i. You can mention, AMQ_HOME/data/kahadb is a storage directory for AMQ messages.
   ii. Active MQ HOME \conf\activemq.xml is the Active MQ configurations file.
   iii. Active MQ HOME\data directory contains the log files:
      - activemq.log
      - audit.log
2. **Apache Tomcat Home**
   i. Apache Tomcat Home\log directory contains automationedge.log file.
   ii. Apache Tomcat Home\webapps directory is used to deploy:
      - aeangine.war
      - aeui.war
   iii. Apache Tomcat Home\webapps\aeui directory has configuration file to set AE UI properties: aeui-config.properties
   iv. Apache Tomcat Home\webapps\aeengine\WEB-INF\classes contains aeengine internal properties files.
      - hibernate.properties
      - log4j2.xml
3. **AutomationEdge Home directory**

   Directory path: <preferred_drive>:/AutomationEdge/aehome

   AutomationEdge Home

   - **workflow**: This folder stores the workflow supporting zip files.
   - **AE_TEMP**: This directory is the temporary storage for temporary files or resources used by AutomationEdge, which are shared across Multiple AE Servers in HA. (Note: If resources are not shared, then a local temp folder should be used instead of one under AE_HOME).
   - **conf**: This folder contains the following configuration files:
     - ae.properties
     - database.properties

- **files**: This folder stores the run time input and output files to workflows
- **psplugins**: This folder contains all the plugins uploaded to AutomationEdge server.

**4. AutomationEdge Enable Debug Logging**

Set the log level values for AE Server, Agent, and Integration Service by modifying log-level details in the respective files.

You need not restart Tomcat service or agent.

**i. AE Server**

Apache Tomcat Home/webapps/aeengine/WEB-INF/classes/log4j2.xml

**ii. Agent**

Agent Home/conf/log4j2.xml

**iii. Integration Service**

Apache Tomcat Home/webapps/aeintegrationservice/WEB-INF/classes/log4j2.xml

After saving the modified files, DEBUG logs appear in the log4j2.xml file.

# Appendix 2: AE installation on CentOS

In the appendix, you will learn about installation on Red Hat Enterprise Linux (RHEL) based Linux distributions, such as CentOS, Fedora, and OEL.

The steps and commands were performed on CentOS7.

## Environment check

Perform an environment check for the following:

- Hostname-IP address resolution
- Firewall settings
- Steps and commands performed in CentOS

## Prerequisites

Install and configure the following software:

- Java
- PostgreSQL with empty **vae** database
- Apache Tomcat
- Apache ActiveMQ

Make necessary changes to set up the CentOS environment.

## AE installation

For AE installation, create a directory structure and database for AE, and deploy Automation engine and UI war files on Tomcat server.

### Create AE directory structure

Create the top-level directory, AE Base Directory in which all AE software and components are installed. You need to create the following directories and setup AE_HOME environment variable:

```
AutomationEdgeBase → <preferred_drive>:/AutomationEdge.

For example, /home/<username>/Automationedge

AutomationEdge Tools → <preferred_drive>:/AutomationEdge/tools.
```

```
For example, /home/<username>/Automationedge/tools
AutomationEdge Home → <preferred_drive>:/AutomationEdge/aehome.
For example, /home/<username>/Automationedge/aehome
```

# Set environment variable

```
AE_HOME= "<path for the AE working directory>".
For example, AE_HOME=/home/<username>/Automationedge/aehome
```

# War files deployment

Following are the steps to deploy war files:

1. Delete all out-of-the-box apps directories from Tomcat, **/tools/apache-tomcat-9.0.0.M8/webapps**.
2. Copy the following .war files in the **/tools/apache-tomcat-9.0.0.M8/webapps** directory.
   - **aeengine.war**
   - **aeui.war**
3. Restart the Tomcat service to extract **.war** files in the **webapps** directory, and database is populated with objects.
4. If Postgres DB user password is different, change it in file **database.properties**.

# Appendix 3: AE installation on Ubuntu

In the appendix, you will learn about installation on Debian based Linux distributions, such as Ubuntu, OpenSuse, and Kali.

The steps and commands were performed on Ubuntu 18.

## Environment check

Perform an environment check for the following:

- Hostname-IP address resolution
- Firewall settings
- Steps and commands performed in Ubuntu 18

## Prerequisites

Install and configure the following software:

- Java
- PostgreSQL with empty **vae** database
- Apache Tomcat
- Apache ActiveMQ

Make necessary changes to set up the Ubuntu environment.

## AE installation

For AE installation, create a directory structure and database for AE, and deploy Automation engine and UI war files on Tomcat server.

### Create AE directory structure

Create the top-level directory, AE Base Directory in which all AE software and components are installed. You need to create the following directories and setup AE_HOME environment variable:

```
AutomationEdgeBase → <preferred_drive>:/AutomationEdge.
For example, /home/<username>/Automationedge
AutomationEdge Tools → <preferred_drive>:/AutomationEdge/tools.
For example, /home/<username>/Automationedge/tools
```

```
AutomationEdge Home → <preferred_drive>:/AutomationEdge/aehome.
For example, /home/<username>/Automationedge/aehome
```

# Set environment variable

```
AE_HOME= "<path for the AE working directory>".
For example, AE_HOME=/home/<username>/Automationedge/aehome
```

# War files deployment

Perform the following steps to deploy war files:

1. Use the following command to change the directory to **webapps**:

   ```
   cd "/home/<user_name>/AE_Tools/apache-tomcat-9.0.5/webapps"
   ```

2. Use the following command to delete all out-of-the-box apps directories or files or folders from the **webapps** directory.

   ```
   rm -rf *
   ```

3. Copy AE war files to the **webapps** directory.

   Use WinSCP tool if you are copying from Windows to Linux.

4. Use following command to rename **aeengine.war** as follows:

   ```
   mv aeengine-4.0.0.0.war aeengine.war
   ```

5. Use following command to rename **aeui.war** as follows:

   ```
   mv aeui-4.0.0.0.war aeui.war
   ```

6. Ensure the following are running:

   i. ActiveMQ: Start using super user credentials.

   ii. PostgreSQL service.

   If Postgres DB user password is different, change it in file /tools/automationedge/conf/database.properties.

   iii. Tomcat: Start using super user credentials.

   Restart the Tomcat service so that .war files get extracted in **webapps** directory, and database is populated with objects.

**==========END OF DOCUMENT==========**