

LICENSE4J AUTO LICENSE GENERATION AND ACTIVATION SERVER USER GUIDE

VERSION 1.5.4



LICENSE4J
www.license4j.com

Table of Contents

Getting Started.....	2
Server Roles	4
Installation	8
Server WAR Deployment	8
Server with Embedded Tomcat Installation	9
Configuration.....	10
Error and Access Logging.....	18
Server License	19

Getting Started

License4J Auto License Generation and Activation Server handles auto license generation requests, activation/deactivation requests, license modification requests, license availability check requests and online license key validation requests.

License4J License Manager GUI tool stores products, templates, and licenses in the defined database server. When a database server other than local embedded derby database is selected in License Manager GUI options, license activation features and product auto license generation settings are enabled.

When an auto license generation request is sent to Auto License Generation and Activation Server, it searches for requested product and template to generate the license. After all security checks performed, it generates the license then sends back as specified in auto license generation output settings for the product.

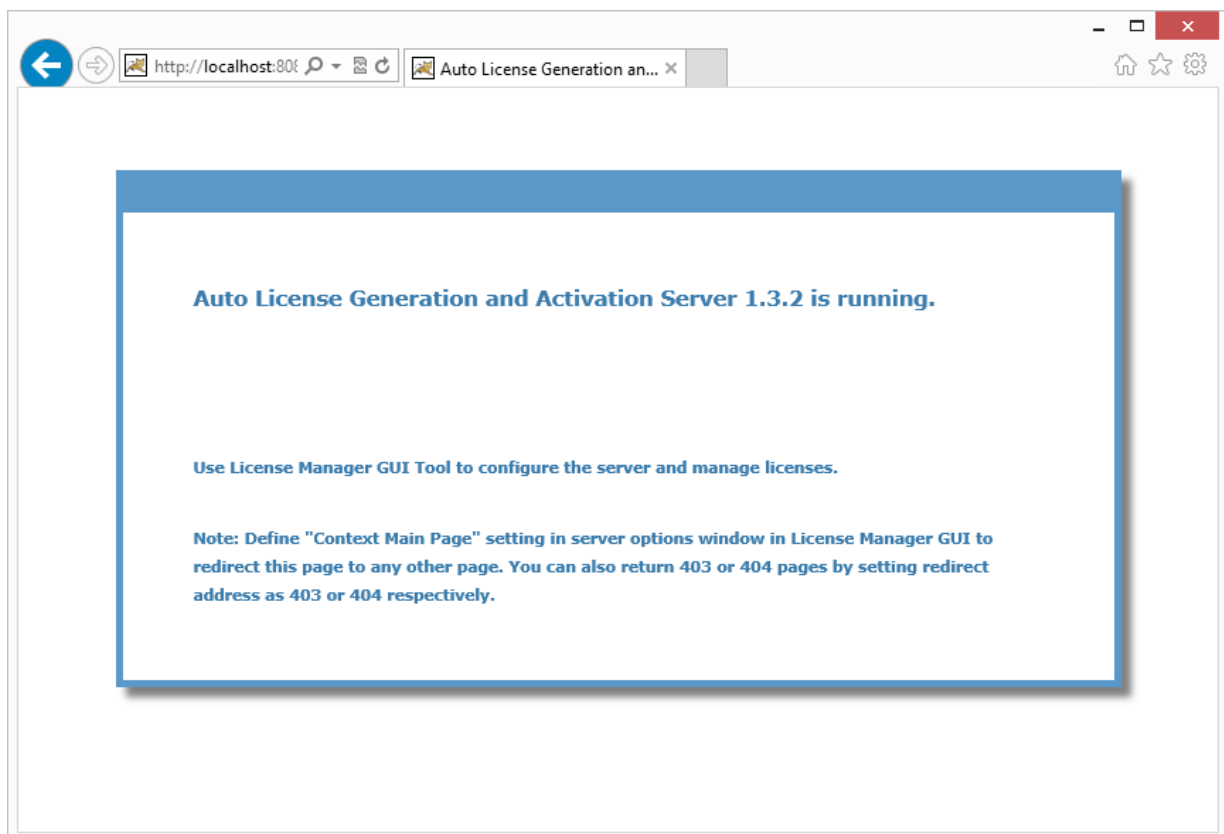
License4J Runtime Library has *autoActivate* method which sends activation request to defined Auto License Generation and Activation Server or Online.License4J system. When an activation request is obtained, requested license is searched in database and if found, allowed activation limit is checked and if activation is allowed, server sends back the activated license text or activation code. If same computer tries to activate license second time, due to e.g. re-installation of software, activation server returns the same license previously activated if hardware ID matches. Therefore when hardware ID does not change re-activation attempts always return the first activation.

License4J defines a new license type which is "Online Basic Key Floating Over Internet". It is very similar to a floating license, but Floating License Server is not required and it works over the Internet. Therefore Auto License Generation and

Activation Server has a role named as "Online Key Validation" for managing online basic keys floating over Internet.

Since License4J Auto License Generation and Activation Server is a web application, it can run on load-balance or fail-over cluster environments.

Auto License Generation and Activation Server does have a management GUI accessible with a browser. The only page that server displays are given in the screenshot below, and License Manager GUI application is the only tool to manage licenses.



Server Roles

License4J Auto License Generation and Activation Server has following ten roles, roles can be enabled and disabled as required. With these roles, more than one server can be deployed and run to balance load on servers or meet security needs.

- **License Generation**

If enabled, server accepts auto license generation requests, and generates licenses. URL will start with http or https depending on server configuration and becomes <http://YourServerName/algas/generate/xxxxxxxx>, the unique number shown as xxxxxxxx after generate keyword is displayed in auto license generation settings window in License Manager GUI. A sample form is given below.

```
<form
action="http://YourServerName/algas/generate/142386147474615165246142386
1451535" method="POST">
    <input type="text" name="name">
    <input type="text" name="email">
    <!-- PUT ALL YOUR INPUT VARIABLES HERE -->
    <input type="submit" value="submit">
</form>
```

The URL in form action can be obtained and form variables should be defined on *Automatic License Generation Settings* window of License4J License Manager. *Automatic License Generation Settings* window has a new menu item as *Display HTML Form Code* to generate license and modification key generation HTML form source.

- **Auto License Activation**

If enabled, server will handle auto license activation requests. The URL for auto activations is <http://YourServerName/algas/autoactivate> and it will accept connections only from Runtime Library *autoActivate* methods.

- **Manual License Activation**

If enabled, manual license activation will be allowed, an html form should be designed for customers to submit activation requests. Form should be submitted to <http://YourServerName/algas/manualactivate> address and should include a hidden value with name *action* and value *activate*; and license activation request string in an input area with name *linfo*. A sample form is given below.

```
<form action="http://YourServerName/algas/manualactivate" method="post">
  <textarea name="linfo"></textarea>
  <input type="hidden" name="action" value="activate">
  <input type="submit" value="submit">
</form>
```

- **Auto License Deactivation**

If enabled, server will handle auto license deactivation requests. The URL for auto deactivations is <http://YourServerName/algas/autodeactivate> and it will accept connections only from Runtime Library *autodeActivate* method.

- **Manual License Deactivation**

If enabled, manual license deactivation will be allowed, an html form should be designed for customers to submit deactivation requests. Form should be submitted to <http://YourServerName/algas/manualdeactivate> and should include a hidden value with name *action* and value *mdeactivate*; and license deactivation request string in an input area with name *linfo*. A sample form is given below.

```
<form action="http://YourServerName/algas/manualdeactivate"
method="post">
  <textarea name="linfo"></textarea>
  <input type="hidden" name="action" value="mdeactivate">
  <input type="submit" value="submit">
</form>
```

- **Manual License Modification**

If enabled, manual license modification will be allowed, an html form should be designed for customers to submit modification request. Form should be submitted to <http://YourServerName/algas/manualmodify> and should include a hidden value with name *action* and value *modify*; and license modification request string in an input area with name *linfo*. A sample form is given below.

```
<form action="http://YourServerName/algas/manualmodify" method="post">
  <textarea name="linfo"></textarea>
  <input type="hidden" name="action" value="modify">
  <input type="submit" value="submit">
</form>
```

- **Online Key Validation**

If enabled, server will lease online basic keys floating over Internet. The URL for license lease is <http://YourServerName/algas/validateobk> and it will accept connections only from Runtime Library *validate* method.

- **License Availability Check**

If enabled, server will respond license availability check requests. Generated licenses or activations can be deleted on server for some reason, and if it is needed to block license owner to use license, it can be checked with a single method in Runtime Library, and if license availability checks fails, software product can be disabled. This role can also be used to blacklist some licenses and detect on customers' computers. It can be used to check for activations. If license deactivation feature is enabled, deactivated license usage can be detected and blocked with availability check. The URL for license check is <http://YourServerName/algas/checkavailability> and it accepts connections only from Runtime Library *validate* method.

Version 1.5.4+ supports license availability check over the web. When a form is submitted with license string (key) and product public key hash value to "isLicenseAvailable" address, server returns 1 if license or activation is available, returns 0 if not. License string should form variable name must be "ls" and public key hash value variable name must be "pa".

```
<form action="http://YourServerName/algas/isLicenseAvailable"
method="post">
    <textarea name="ls"></textarea>
    <input type="hidden" name="pa" value="public key hash value">
    <input type="submit" value="submit">
</form>
```

- **License Modification**

If enabled, server will allow auto license modification requests. The URL for license modification is <http://YourServerName/algas/autoactivate> and it will accept connections only from Runtime Library *modifyLicense* method. License modification is available for only activated licenses with activation return method as license text. Activated license text is modified if given modification key is valid, and new modified activated license text is returned to user.

- **Product Update Notification and License Message Role**

Product update notifications and license messages can be defined for any product and license. There are required Runtime library methods to check for update and/or license message. The URL for this role is <http://YourServerName/algas/checkmessageupdate> and it will accept connections only from Runtime Library methods *checkForUpdate* and *checkForNewMessage* methods.

Version 1.5.0 and above also accepts all requests on <http://YourServerName/algas/> URL except for license generation. Therefore license server URL can be defined

same for all runtime methods. "algas" is the context name, a different context name is used, it should be replaced.

Auto License Generation and Activation Server install with embedded Tomcat also accepts requests on all URLs, there is no specific context for application. Therefore it is possible to use just the server name without any path like <http://YourServerName/>

In addition to these ten server roles License4J Auto License Generation and Activation Server also runs a time servlet to provide web time to License4J Runtime Library to check date/time if a license has enabled online date/time check feature. The time source URL is <http://YourServerName/algas/time> and can be defined in online date/time check feature web server field in license generation wizard of License Manager GUI.

Installation

Server WAR Deployment

Auto License Generation and Activation Server is tested on Apache Tomcat version 7 and version 8 with JDK 1.7 and 1.8. All required libraries are included in application WAR file.

Download package includes the application deployment file (WAR file) and a sample configuration file. Copying these two files is sufficient to complete installation.

1. Copy *license4j-server.conf* file to Tomcat configuration directory. Directory name is *conf* and can be found in Tomcat installation path.
2. Open *license4j-server.conf* file and define the only required keyword **jdbc-url** for your database which you connect with License Manager GUI and generate licenses (MySQL, PostgreSQL or MS SQL Server is required).

3. Copy *algas.war* file to Tomcat *webapps* folder for auto deployment.
4. Copy your database server JDBC driver jar file to Tomcat lib directory.
5. If Tomcat is already started, it will be deployed. If Tomcat is not running, start Tomcat.
6. When application is deployed, it connects to Online.License4J and activates included trial license. After first startup it can be changed on Auto License Generation and Activation server settings window of License Manager GUI.

A log file is created in Tomcat logs directory with name *license4-server.log*, check the log file for errors and information messages. An access log file (*license4-access.log*) is created in the same directory and includes all failed and successful requests.

Server with Embedded Tomcat Installation

Auto License Generation and Activation Server with embedded Tomcat installation includes both an embedded Tomcat and Java 1.8. Setup file is distributed for Windows and Linux operating systems and installs all required files on operating system; then a sample configuration file is created on first run.

After installation completed, installed service is started and default configuration file is created, required ***jdbc-url*** keyword must be defined in configuration file. By default server listens on port 8090, you can browse to <http://localhost:8090> and see the server status. If server is not running due to an error in configuration file or database, the main reason for error is displayed on the page.

Configuration file and logs files are located in local *ProgramData* folder (C:\ProgramData\ALGAS) on Windows, and in installation path on Linux.

Created default configuration file has required descriptions for configuration keywords. Different from WAR file deployment, you can define interface and port number on which server will listen, and SSL certificate settings to accept secure connections.

Log file is created in program data folder (C:\ProgramData\ALGAS) on Windows and in installation path on Linux. An access log file (*license4-access.log*) is created in the same directory and includes all failed and successful requests.

Configuration

License4J Auto License Generation and Activation Server needs only *jdbc-url* keyword to be specified in a configuration file. The configuration file name is *license4j-server.conf* and should be placed in Tomcat configuration directory named as *conf*, and in *ProgramData* (C:\ProgramData\ALGAS) folder on Windows and installation folder on Linux. A sample configuration file is included in download package which includes sample jdbc url definitions for MySQL, Postgresql and MS SQL database servers.

Version 1.4.2+ has additional configuration settings for JDBC connection pooling and Tomcat threads. JDBC connection pool settings can be configured if more database connections are required. Number of minimum and maximum Tomcat threads with keep alive settings gives more control when embedded server is used. More information and explanations are given in comment blocks in default configuration file.

License4J License Manager GUI is used to configure Auto License Generation and Activation Server. *Server settings* menu item is found in *Tools* menu. As in the following screenshot, all server settings are displayed and can be modified.

License4J Auto License Generation and Activation Server is scheduled to check for configuration changes each 10 seconds so after changing any setting, it is not required to restart or redeploy application.

Auto License Generation and Activation Server Options

Auto License Generation and Activation Servers

Server List:

Server Information

Name:

Version: Hostname:

Description:

Hardware ID:

Tomcat Path:

Context Main Page:

License Key:

Maintenance End:

Server Roles

<input checked="" type="checkbox"/> License Generation	<input checked="" type="checkbox"/> Online Key Validation	<input checked="" type="checkbox"/> License Availability Check
<input checked="" type="checkbox"/> Auto License Activation	<input checked="" type="checkbox"/> Auto License Deactivation	<input checked="" type="checkbox"/> Auto License Modification
<input checked="" type="checkbox"/> Manual License Activation	<input checked="" type="checkbox"/> Manual License Deactivation	<input checked="" type="checkbox"/> Manual License Modification

Name and Description fields are optional and used just for displaying more meaningful information. Server's Hardware ID and license key is displayed and you

can change license key with the related button. Context main page setting defines the address which <http://YourServerName/algas/> main application context page will be redirected. If main context page will be redirected to another page address should be given as <http://www.example.com>; if it is required to return a 403 or 404 page, 403 or 404 should be set relatively. e.g. When 403 is given as context main page, <http://YourServerName/algas/> page will return a 403 forbidden page. *Remove Server* button removes only server settings from database, if removed server is still running it is stopped. When tomcat is restarted on which a removed Auto License Generation and Activation Server deployed, server settings is reset to default trial settings and created on the database.

Change Key button displays a dialog to enter your license key for Auto License Generation and Activation Server. *Enter Modification Key* button is used to enter a modification key to modify current license. It is used to extend maintenance period when a maintenance is purchased.

Auto License Generation and Activation Server sends notification e-mails for selected cases. It can send notifications for every successful and failed request server gets. User notification e-mails can be customized with e-mail templates. There are defined strings between curly braces which are replaced with values from licenses and products.

SMTP server information must be defined for server to connect and send e-mails. It supports TLS connections and SMTP authentication. Values in from fields are used in all e-mails sent.

The screenshot shows a 'Server Notification Settings' window with the following elements:

- Tabbed Interface:** Three tabs are visible: 'SMTP Server Information' (active), 'License Action Notifications', and 'License Expire Notifications'.
- SMTP Server Information Tab:**
 - SMTP Host:** A text input field.
 - Port Number:** A text input field containing the value '25'.
 - From Name:** A text input field with a tooltip '(Display name in sent e-mails)'.
 - From Address:** A text input field with a tooltip '(Single From e-mail address)'.
 - To Address:** A text input field with a tooltip '(Server Admins, semicolon separated)'.
 - Use TLS:** An unchecked checkbox.
 - Require Authentication:** An unchecked checkbox.
 - Username:** A text input field.
 - Password:** A text input field.
- Bottom Buttons:**
 - 'Test SMTP Settings' (disabled)
 - 'User Notification e-mail Templates' (disabled)
 - 'Save' (active)
 - 'Cancel' (disabled)

License Action Notifications tab as in the screenshot below is used to define actions for which to send e-mails. Server admin notification e-mails are sent to defined e-mail addresses on first tab as "From Address". They are used as server administrators. User notification e-mails are sent to user e-mail address in license. If an e-mail address is not defined in license, user e-mails cannot be sent.

User notifications for some cases are also not possible when license string sent by runtime library cannot be found on license server database.

	Server Admin Notifications		User Notifications	
	Success	Fail	Success	Fail
Auto License Generation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto Modification Key Generation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto License Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manual License Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto License Re-Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manual License Re-Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto License Deactivation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manual License Deactivation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto License Modification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manual License Modification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Online Key Validation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
License Availability Check	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Only one license expire notification is sent for each license if notifications enabled. Notifications are sent either after license or maintenance expired or defined days left to expiration. More than one value can be defined for "Days to License Expire" fields; numbers should be separated with semicolon. e.g. if defined values are 1,7,14 then expire notifications are sent before 14 days, 7 days and 1 day of expiration date.

Server Notification Settings

SMTP Server Information License Action Notifications **License Expire Notifications**

	Server Admin Notifications	User Notifications
Days to License Expire <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Days to Maintenance Expire <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

(Comma seperated number of days should be defined to send notification e-mails before expiration.)
(e.g. 1,7,14 to send notification e-mails 1 day, 7 days and 14 days before.)

After License Expired	<input type="checkbox"/>	<input type="checkbox"/>
After Maintenance Expired	<input type="checkbox"/>	<input type="checkbox"/>

(One notification e-mail is sent when expired.)

User Notification e-mail Templates Save Cancel

There are default user notification e-mail templates available on database; they can be edited in e-mail template editor. Combo box lists all actions to edit. Some strings between curly braces are replaced with values from license and product.

User Notification e-mail Templates

Template Selection

Template: Successful Auto License Activation

Edit e-mail Template

CC:

BCC:

Subject:

☐ Send License String Attached with File Name:

Basic Editor HTML Source

Message:

```

Product Name: {Product-Name}
License ID: {License-ID}
Activation ID: {Activation-ID}
Registered To: {User-RegisteredTo}
Company: {User-Company}
Activation count out of total allowed: {Activation-TotalCount}/{Activation-AllowedCount}

Regards
    
```

Reset To Default Save Close

String between curly braces below are replaced by server while sending e-mails.

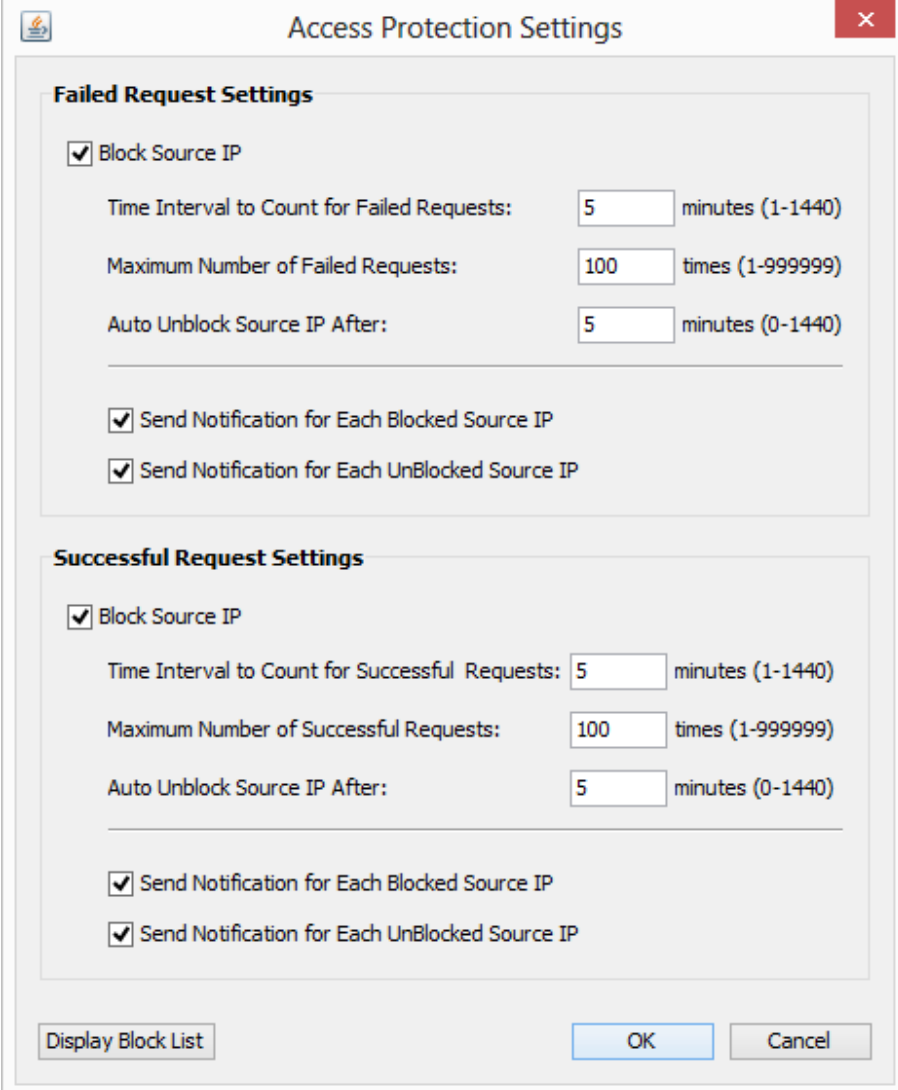
{Server-Admins}	Administrator e-mails defined on SMTP information tab.
{Request-ResultMessage}	The resulting message either successful or fail.
{Request-FromIP}	IP address which request sent from.
{Request-FromHostName}	Hostname request sent from.
{Request-FromOS}	OS name request sent from.
{Product-ID}	Unique product ID.
{Product-Name}	Defined product name.
{Product-Version}	Product version if defined.
{Product-Edition}	Product edition if defined.
{License-ID}	Unique license id.
{License-LicenseString}	License string either key or long license text.
{Activation-TotalCount}	Activation number, how many activation are made.
{Activation-AllowedCount}	Allowed activation count for license.
{Deactivation-AllowedCount}	Allowed deactivation count for license.
{Deactivation-TotalCount}	Deactivation count.
{License-ValidityExpiration}	Expiration date in format yyyy-MM-dd HH:mm:ss
{License-MaintenanceExpiration}	Maintenance expiration date in format yyyy-MM-dd HH:mm:ss
{License-Quantity}	License quantity.
{License-HardwareID}	Hardware ID if license is node-locked.
{Activation-ID}	Unique activation ID.
{Activation-LicenseString}	Activation string (activated license string)
{Activation-HostName}	Hostname on which activation is completed.

{Activation-FromIP}	IP address which activation completed.
{Activation-HardwareID}	Hardware ID used in activation.
{Activation-ValidityExpiration}	Expiration date for activation (if “set activated license text generation time to activation time” is selected for license).
{Activation-MaintenanceExpiration}	Maintenance expiration date for activation (if “set activated license text generation time to activation time” is selected for license).
{ModificationKey-ID}	Unique modification key ID.
{ModificationKey-LicenseString}	Modification key itself.
{ModificationKey-DedicatedLicenseID}	License ID if modification key is dedicated to.
{ModificationKey-AllowedUsageCount}	Allowed use count for modification key.
{User-FullName}	User name defined in license.
{User-EMail}	User e-mail address in license.
{User-RegisteredTo}	Registered To field in license.
{User-Company}	Company name in license.
{User-Telephone}	Telephone number in license.
{User-City}	City in license.

Access protection settings can be defined for server so it can block requesting IPs if defined limits exceeded. Settings and limits for failed requests and successful requests are defined separately. “Time Interval to Count for Failed/Successful Requests” defines time interval to save all failed and successful requests to database. If a single IP address access count exceeds the maximum number of failed/successful access count, then the source IP is added to the block list and all new requests are blocked for a duration of value defined in “Auto Unblock Source IP After” field. If the defined value in “Auto Unblock Source IP After” is zero, then the blocked IP address is never removed from the list automatically, but it can be removed manually. The current block list can be viewed with “Display Block List” button.

If e-mail server and recipients are defined in notification settings window, then all block and unblock notifications can be send to defined recipients.

In the screenshot below, if server receives 100 failed or successful requests from same IP address in last 5 minutes, it blocks IP address for 5 minutes. After 5 minutes, it removes IP address from block list.



The image shows a Windows-style dialog box titled "Access Protection Settings". It has a standard icon in the top-left corner and a red close button in the top-right corner. The dialog is divided into two main sections: "Failed Request Settings" and "Successful Request Settings". Each section contains a "Block Source IP" checkbox, which is checked in both. Below each checkbox are three input fields: "Time Interval to Count for Failed/Successful Requests" (set to 5 minutes), "Maximum Number of Failed/Successful Requests" (set to 100 times), and "Auto Unblock Source IP After" (set to 5 minutes). At the bottom of each section are two more checkboxes: "Send Notification for Each Blocked Source IP" and "Send Notification for Each UnBlocked Source IP", both of which are also checked. At the bottom of the dialog, there is a "Display Block List" button on the left, and "OK" and "Cancel" buttons on the right.

Access Protection Settings

Failed Request Settings

- ☒ Block Source IP
 - Time Interval to Count for Failed Requests: 5 minutes (1-1440)
 - Maximum Number of Failed Requests: 100 times (1-999999)
 - Auto Unblock Source IP After: 5 minutes (0-1440)
- ☒ Send Notification for Each Blocked Source IP
- ☒ Send Notification for Each UnBlocked Source IP

Successful Request Settings

- ☒ Block Source IP
 - Time Interval to Count for Successful Requests: 5 minutes (1-1440)
 - Maximum Number of Successful Requests: 100 times (1-999999)
 - Auto Unblock Source IP After: 5 minutes (0-1440)
- ☒ Send Notification for Each Blocked Source IP
- ☒ Send Notification for Each UnBlocked Source IP

Display Block List OK Cancel

Error and Access Logging

Server stores access and error log files in a folder names as **logs** in *ProgramData* folder (*C:\ProgramData\ALGAS\logs*) on Windows and in installation directory on Linux. It keeps two log files: *license4j-server.log* file includes server messages and errors; *license4j-access.log* file includes access messages.

Server License

One license is required for each computer running License4J Auto License Generation and Activation Server. Download package includes a 30 days valid trial license, and it activates trial license on first run.

When a license is purchased, new license key should be set in server settings window. Given license key will be auto activated on first run.

End of document.