

JetBrains License Server User Guide

- Introduction
 - About JetBrains License Server
 - System Requirements
 - Hardware
 - Software
 - Distribution Options
- Installing and Configuring License Server
 - Installing License Server
 - Configuring log files location
 - Configuring Automatic Server Discovery
 - Configuring License Server to Use an External Database
 - Extending License Server to Use Custom Verification
 - How to Provide Custom Verification
 - Sample Verification Plug-in
- Logging In
 - Launching License Server Control Panel
 - Setup Server
 - Login
- License Server Settings
 - Settings
 - Status
- Product-Specific Control Panel
 - General
 - License Keys
 - Permanent Tickets
 - Floating Tickets
 - Report
 - Settings
 - Log

Introduction

About JetBrains License Server

JetBrains License Server is a web application that enables license administration across a single network. License Server serves as a central point for distribution of licenses among multiple users and client machines in a network environment. It supports the following JetBrains products:

- IntelliJ IDEA 7.0 or higher (only Commercial licenses). (A License Server-compatible build of IntelliJ IDEA 6.0 is available upon [request](#).)
- ReSharper 3.1 or higher (only Commercial licenses).
- ReSharper Cpp 1.0 or higher
- dotTrace 5.5 or higher (only Commercial licenses). (Floating licenses are supported starting from dotTrace 3.1).
- dotMemory 4.0 or higher (only Commercial licenses).
- dotCover 1.0 or higher.
- RubyMine 1.0 or higher.
- PyCharm 1.0 or higher.
- WebStorm 1.0 or higher
- PhpStorm 1.0 or higher
- AppCode 1.0 or higher
- CLion 1.0 or higher

License Server issues and revokes license tickets to/from network clients based on the license keys that are provided by JetBrains after purchase. The License Server allows for single-user keys to be used as concurrent licenses. The License Server will allow for the exact number of concurrent instances as purchased commercial licenses imported into the License Server.

That said, each license key provides one ticket. A single ticket grants permission to use a single copy of a product. License Server receives requests for license tickets from client applications and issues tickets to them upon verification, eliminating the need to configure clients individually.

There are two kinds of license tickets:

- Floating tickets are issued for a limited period of time and prolonged on a regular basis.
- Permanent tickets are issued without a specific time limit. They are released manually by a client application or server administrator.

System Requirements

Hardware

- 256 MB RAM

Software

- Java SE Runtime Environment (JRE)/Java SE Development Kit (JDK) 5 or higher installed in any of the following operating systems:
 - Windows® (2000, NT 4.0 or higher).
 - Linux/UNIX-based OS.
 - Macintosh OS X™.
- Apache Tomcat 6.0.
- Mozilla Firefox 2.0+, Internet Explorer 7.0+, or Safari 3.0+ for accessing the License Server control panel.

Distribution Options

License Server is supplied in two distribution options:

- As a standalone WAR file for deploying in an application server.
- As a package that includes Apache Tomcat.

Installing and Configuring License Server

Installing License Server

To install and configure License Server bundled with Apache Tomcat:

1. Install JRE or JDK:
 - a. Download and install Java SE Runtime Environment (JRE) 5 or later.
 - b. Create an environment variable `JAVA_HOME` and assign it to JRE installation home using forward slashes in the path (for example, `c:/Program Files/Java/jre1.6.0_03` or `/usr/local/java/jre`):
 - To define `JAVA_HOME` environment variable under Windows XP:
 - i. Select Start | Settings | Control Panel | System | Advanced | Environment Variables.
 - ii. In the User environment variables for <user name> area, click Create.
 - iii. In the Variable name field of the New System Variable dialog box, enter `JAVA_HOME`.
 - iv. In the Variable value field, enter the path to JRE installation home.
 - v. Click OK.
 - To define `JAVA_HOME` environment variable under UNIX/Linux, execute the following command:

```
env JAVA_HOME=path
```

where path should reference JRE installation home.
2. Start bundled Apache Tomcat.

To start Apache Tomcat distribution bundled with License Server and deploy `licenseServer.war` package, execute one of the following scripts:

 - `<Tomcat root>\bin\startup.bat` (Windows).
 - `<Tomcat root>/bin/startup.sh` (UNIX/Linux/MacOS).

License Server will start listening on port 8080 with `licenseServer` path prefix.
3. Start License Server by typing `http://<host-name>:8080/licenseServer` in the address bar of your web browser.

To deploy License Server standalone WAR file under Apache Tomcat:

1. Copy `licenseServer.war` from `<delivery package>/apache-tomcat-<version>/webapps/` into `<Tomcat root>/webapps` subdirectory.
2. Restart Apache Tomcat.
3. Start License Server by typing `http://<host-name>:<port-number>/licenseServer` in the address bar of your web browser.
4. Proceed with setting up your License Server.

For either License Server distribution, you can optionally do any of the following:

- [Configure automatic License Server discovery](#)
- [Configure License Server to use an external database](#)

Configuring log files location

By default license server's log files are written into `<Tomcat-Home>/logs/jetbrains-license-server` directory.

The logs location can be changed using `jetbrains.license.server.logs` Java property.

For Tomcat it can be done via the `JAVA_OPTS` environment variable. Add a new system environment variable `JAVA_OPTS` with the value like this:

```
-Djetbrains.license.server.logs=c:/custom/logs/folder
```

Note that system reboot may be necessary for the environment variables to take effect.

Configuring Automatic Server Discovery

To configure automatic License Server discovery:

Add a DNS TXT record `url=<server_url>` for the following name:

```
_jetbrains-license-server.<network-domain-name>
```

To verify the record in a Unix environment:

Run the following command:

```
dig _jetbrains-license-server.<domain_name> TXT
```

A valid response should look like this:

```
_jetbrains-license-server.acme.com. 3600 IN TXT "url=http://lserver:8080/licenseServer"
```

Configuring License Server to Use an External Database

License Server comes with embedded Apache Derby database. However, you can configure License Server to work with an external database. The following databases are supported:

- MySQL 5
- Sybase Adaptive Server Enterprise (ASE) 15

To migrate to an external database, make the following modifications to `licenseServer.war/WEB-INF/classes/META-INF/mod-elContext.xml`:

1. In `org.apache.commons.dbcp.BasicDataSource` bean properties, comment out `driverClassName` property referencing the embedded database, and uncomment `driverClassName` property corresponding to the external database of your choice.
2. In `com.jetbrains.licenseServer.model.impl.LSTopLinkJpaVendorAdapter` bean properties, comment out `databasePlatform` property that references `oracle.toplink.essentials.platform.database.DerbyPlatform` SQL dialect, and uncomment the `databasePlatform` property corresponding to the external database of your choice.
3. Replace default values for `url`, `username`, and `password` properties of `dataSource` bean with production values. Make sure to set necessary database connection settings using the `url` property.

Extending License Server to Use Custom Verification

You can extend License Server to verify clients in one or more ways before they can obtain tickets. When you add one or more custom verifications, the following rules apply:

- If a user requesting a license passes all verifications, he or she receives a license ticket.
- If a user fails to pass at least one verification, his/her ticket request is rejected.

You can add as many verifications as required by your corporate policy. All your verifications are executed one by one during request processing.

License Server doesn't execute custom verifications in a particular order. It is your responsibility to develop verification rules the way that doesn't depend on the order in which they are executed.

License Server doesn't cache verification results, meaning that a client is verified for each request it sends.

How to Provide Custom Verification

To apply a custom verification procedure, you should create a JavaBean implementing `ClientVerifier` public interface:

ClientVerifier.java

```
/**
 * @param productFamilyId - Product family ID
 * @param userName      - User name how it comes in a ticket request
 *                       from a product, i.e. IntelliJ IDEA, ReSharper, etc.
 * @param hostName      - Host name how it comes in a ticket request
 *                       from a product, i.e. IntelliJ IDEA, ReSharper etc.
 *                       In the log file, user ID is represented as userName@hostName
 * @param machineId     - User machine-specific ID
 * @return TRUE if a request with the specified userName, hostName and machineId for a specified
 product
 *                should be accepted as a result of successful authorization,
 *                and FALSE otherwise
 */

public interface ClientVerifier {
    boolean isAuthorized(String productFamilyId, String userName, String hostName, String
 machineId);
}
```

If `isAuthorized` method in any `ClientVerifier` implementation returns false, the requesting client is considered unauthorized and is not granted a license ticket.

For every implementation of `ClientVerifier`, you should create a separate Spring bean. Every such bean should be added to License Server classpath (`<Tomcat version>/webapps/licenseServer/WEB-INF/classes/`) and registered as a standard Spring bean using a bean descriptor in an existing application context file or in a new file named `classpath/META-INF/<pluginName>-plugin.xml`, such as the following:

yourCustomPlugin-plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">
  <bean id="yourCustomPlugin" class="A.B.C.D"/>
</beans>
```

To sum it up, a registered License Server verification plug-in consists of:

1. Plug-in bean definition: `licenseServerClasspath/META-INF/<pluginName>-plugin.xml`
2. Actual bean in License Server classpath.

Sample Verification Plug-in

The following is a sample implementation of `CustomVerifier` interface:

ClientVerifierExample.java

```
package com.intellij.licenseServer.plugin.nameVerifier;

import com.jetbrains.licenseServer.openapi.ClientVerifier;

import java.util.List;

/**
 * This is a simple implementation of ClientVerifier
 * If Type==BLACK_LIST - if a user name specified in a ticket request is found in a black list,
 * the request is rejected.
 * If Type==WHITE_LIST - if a user name specified in a ticket request is found in a white list,
 * the request is accepted.
 *
 * @author shkate@jetbrains.com
 */
public class ClientVerifierExample implements ClientVerifier {

    public enum Type {
        BLACK_LIST, WHITE_LIST
    }

    private Type type = Type.BLACK_LIST;

    private List<String> userNames;

    public void setType(Type type) {
        this.type = type;
    }

    public void setUserNames(List<String> userNames) {
        this.userNames = userNames;
    }

    public boolean isAuthorized(String productFamilyId, String userName, String hostName, String
machineId) {
        if (userName == null) {
            return type != Type.WHITE_LIST;
        }
        return (type == Type.BLACK_LIST && !userNames.contains(userName)) || (type ==
Type.WHITE_LIST && userNames.contains(userName));
    }
}
```

Here's a bean descriptor used to register the sample custom verification plug-in in License Server:

META-INF/verifier-plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-2.0.xsd">
  <bean id="baseNameVerifier"
class="com.intelij.licenseServer.plugin.nameVerifier.ClientVerifierExample">
    <property name="type" value="BLACK_LIST"/>
    <property name="userNames">
      <util:list>
        <value>user1</value>
      </util:list>
    </property>
  </bean>
</beans>
```

Logging In

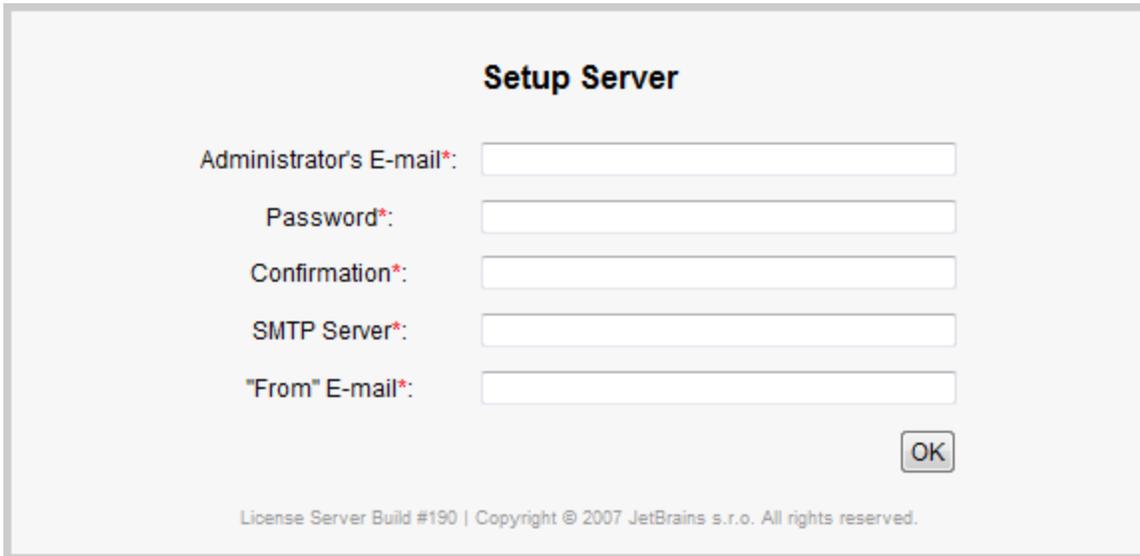
Launching License Server Control Panel

Open your web browser and type `http://<host-name>:8080/licenseServer` in the address bar.

Setup Server

When you start License Server in your web browser for the first time, the Setup Server page opens (fig. 1) where you should complete four mandatory fields:

- Administrator's E-mail (used as admin ID.)
- Password (min. 6 characters.)
- Confirmation (confirm password.)
- SMTP Server (the address of your company's SMTP server.)
- "From" address (e-mail address to be used as a sender of any License Server messages.)



Setup Server

Administrator's E-mail*:

Password*:

Confirmation*:

SMTP Server*:

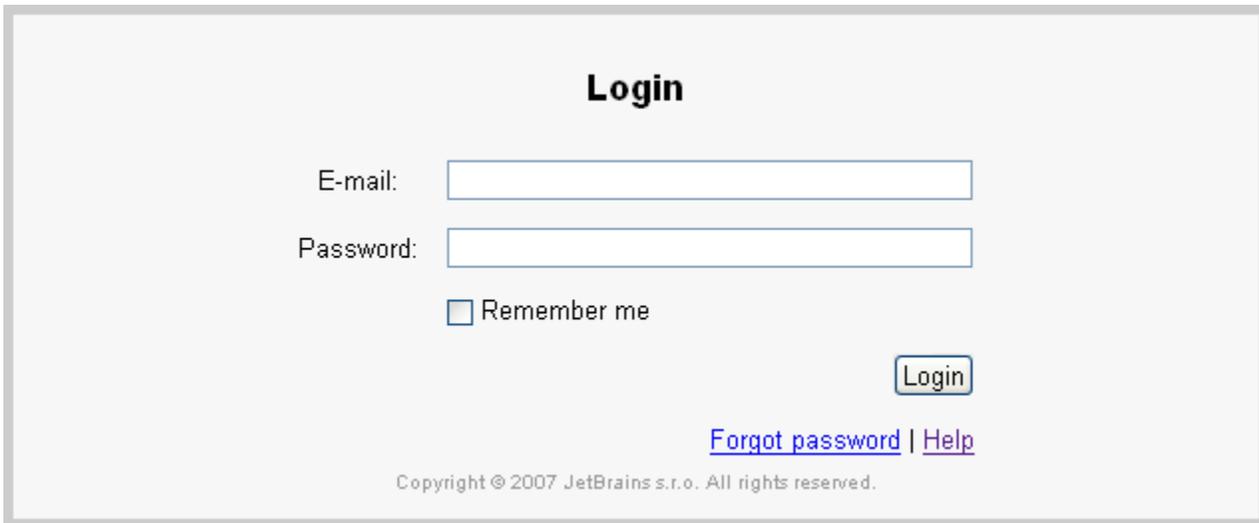
"From" E-mail*:

License Server Build #190 | Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 1. The Setup Server page

Login

Every time you open License Server after you have set it up, the Login page displays (fig. 2).



Login

E-mail:

Password:

Remember me

[Forgot password](#) | [Help](#)

Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 2. The Login page

Enter the credentials that you specified during server setup in the E-mail and Password fields. Click Remember me so that License Server recognizes you at any time, unless the system is rebooted. You can open other web resources or close the browser window in the meantime. You can subsequently log out by clicking the Logout link in the top right corner of any License Server page.

License Server Settings

After you have successfully logged in, the JetBrains License Server home page displays (fig. 3). It consists of two tabs, Status and Settings.

JetBrains License Server [Logout](#) | [Help](#)

Status Settings

License server is up, running and receiving requests.

Product Statistics

| Product | Total Keys | Total Tickets | Free Tickets | Issued Tickets |
|-------------------------------|------------|---------------|--------------|----------------|
| IntelliJ IDEA | 1 | ∞ | ∞ | 0 |
| ReSharper | 1 | 1 | 0 | 1 |
| RubyMine | 0 | 0 | 0 | 0 |
| dotTrace | 0 | 0 | 0 | 0 |

License Server Build #201 | Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 3. JetBrains License Server | Status

Settings

The Settings page (fig. 4) contains a number of essential server settings that can be modified when necessary:

- Administrator's E-mail: the e-mail address that you entered during server setup and used to log in.
- Password: administrator's password.
- Confirmation: password confirmation.
- SMTP Server: the address of your SMTP server that is used to issue permanent licenses (specifically, to send and receive activation codes.)
- "From" address: e-mail address to be used as a sender of any License Server messages.

JetBrains License Server [Logout](#) | [Help](#)

Status Settings

Server Settings

Administrator's E-mail*:

Password*:

Confirmation*:

SMTP Server*:

"From" E-mail*:

License Server Build #190 | Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 4. JetBrains License Server | Settings

Status

The Status page (see [fig. 3](#)) contains a table designed to track ticket allocation statistics server-wide, across all supported products.

You can open a detailed product-specific control panel by clicking the name of the corresponding product (for example, IntelliJ IDEA) in this table.

Product-Specific Control Panel

A product-specific control panel (for example, IntelliJ IDEA) includes seven tabs:

- General
- License Keys
- Permanent Tickets
- Floating Tickets
- Report
- Settings
- Log

General

The General tab ([fig. 5](#)) contains a product-specific statistics table with four rows:

- License Keys: number of license keys that you have registered with the server.
- Total Tickets: total number of tickets provided by registered keys.
- Free Tickets: number of tickets available for issuing.
- Issued Tickets: number of issued tickets.



The screenshot shows the 'JetBrains License Server' interface for 'IntelliJ IDEA'. At the top right is a 'Logout' link. Below the header is a navigation bar with seven tabs: 'General' (selected), 'License Keys', 'Permanent Tickets', 'Floating Tickets', 'Report', 'Settings', and 'Log'. The main content area is titled 'Statistics' and contains a table with the following data:

| | |
|----------------|----|
| License Keys | 10 |
| Total Tickets | ∞ |
| Free Tickets | ∞ |
| Issued Tickets | 1 |

At the bottom of the page, there is a copyright notice: 'Copyright © 2007 JetBrains s.r.o. All rights reserved.'

Figure 5. Product-Specific Control Panel | General

If you have added an unlimited license key, Total Tickets and Free Tickets rows are not assigned a numerical value because you can issue as many tickets as you wish.

This table is only populated with non-zero values after you have added at least one license key.

License Keys

This tab contains the Add Keys From Purchase E-mail link. Click it to open a pop-up window ([fig. 6](#)), and paste the entire body

of the e-mail message with license keys provided to you by a JetBrains representative.

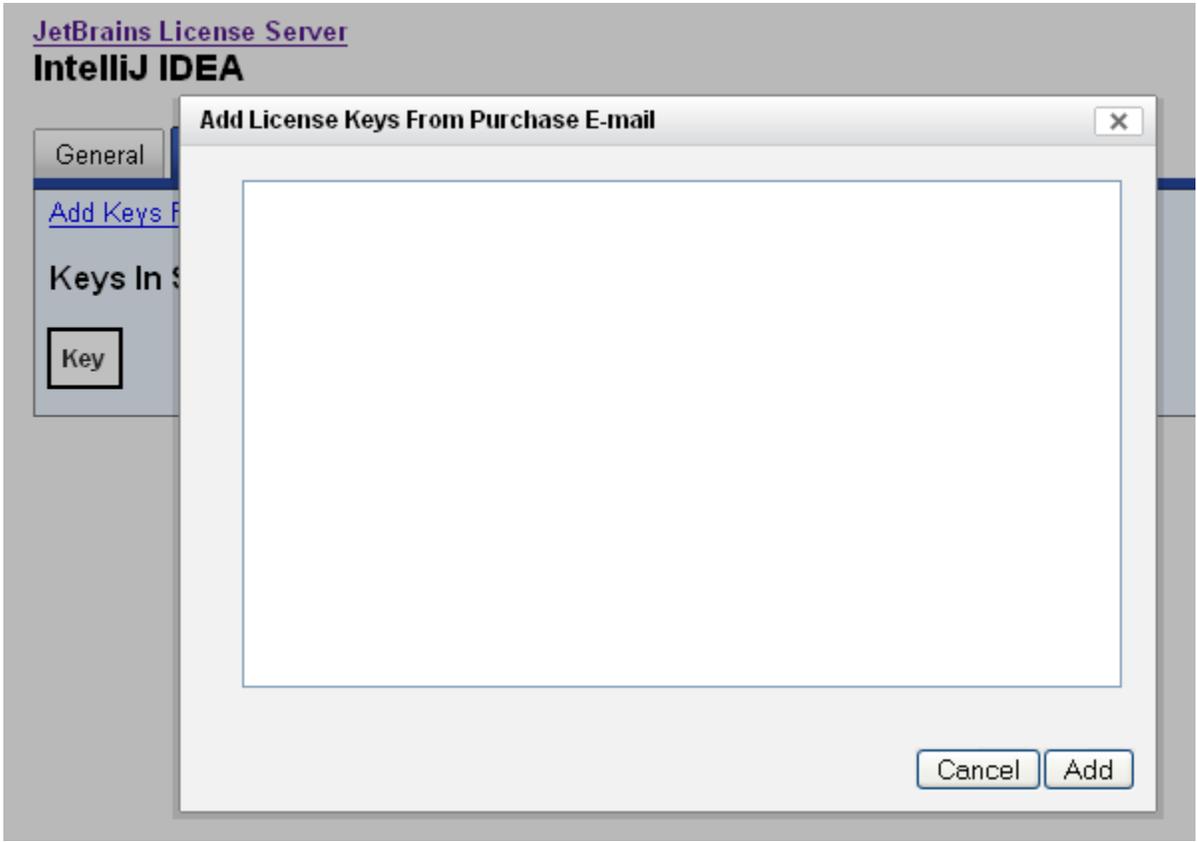


Figure 6. Product-Specific Control Panel | License Keys | Add License Keys From Purchase E-Mail

After you click Add, this window closes and the number of keys that were processed and saved displays in the yellow box at the top of the License Keys tab (fig. 7).

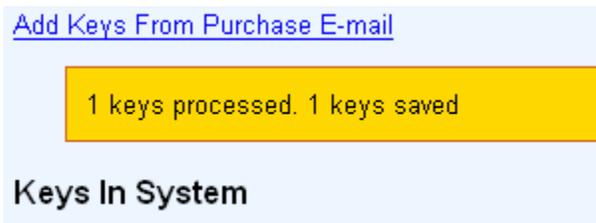


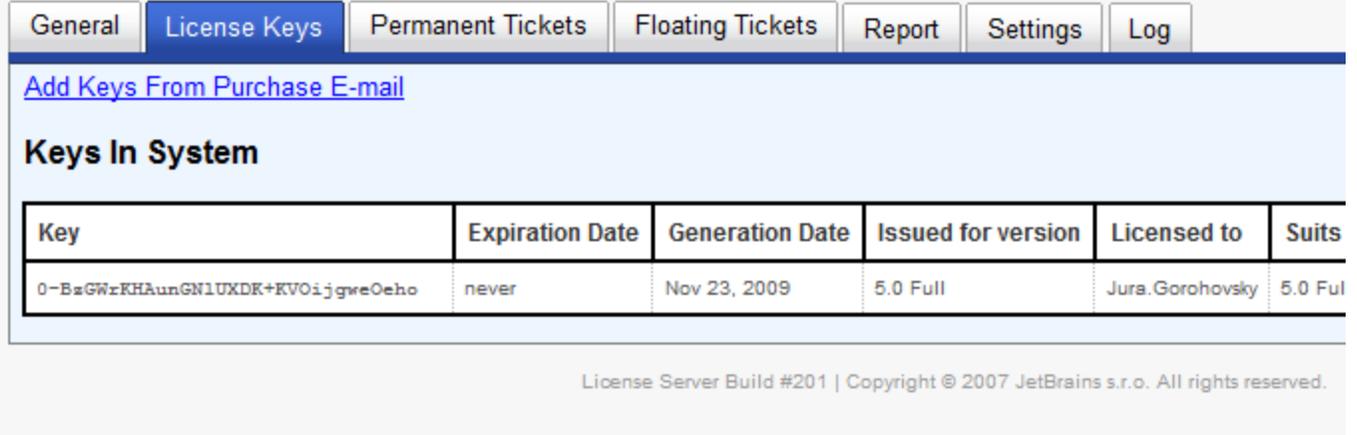
Figure 7. The message that displays after processing and saving license keys

If no keys were processed and saved after you've copied the message into the pop-up window, make sure you've pasted the entire message body.

If the controls in Add License Keys From Purchase E-mail are grayed out, try upgrading your browser to Internet Explorer 7.0 or Mozilla Firefox.

After you have added at least one key, the Keys In System table displays in the License Keys tab (fig. 8).

JetBrains License Server ReSharper



General License Keys Permanent Tickets Floating Tickets Report Settings Log

[Add Keys From Purchase E-mail](#)

Keys In System

| Key | Expiration Date | Generation Date | Issued for version | Licensed to | Suits |
|----------------------------------|-----------------|-----------------|--------------------|-----------------|---------|
| 0-BzGWzKHAunGN1UXDK+KVOijgweOeho | never | Nov 23, 2009 | 5.0 Full | Jura.Gorohovsky | 5.0 Ful |

License Server Build #201 | Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 8. Product-Specific Control Panel | License Keys | Keys In System

This table contains the following columns:

- Key: one or more license keys provided by JetBrains.
- Expiration date: expiration date of the license key.
- Generation date: date of key generation.
- Issued for version: product version that a key was generated for.
- Licensed to: the licensee's name.
- Suits to version: version of the product that this license key makes available.
- <Last column>: allows deleting a previously added key.

Any license key provided for a certain version of a product makes available not only this version, but all of its preceding versions. For example, a license key generated for IntelliJ IDEA 7.0 allows you to use IDEA 7.0, IDEA 6, and all other legacy versions of IDEA.

License keys and server settings are stored in an encrypted database located at <Apache Tomcat root>/db/. You can migrate to an external database if required.

Permanent Tickets

This tab displays only if the Enable Permanent Tickets check box is selected in a product-specific Settings tab. It contains a table that identifies clients who have requested and received permanent tickets.

- ✔ To obtain a permanent ticket for a copy of IntelliJ IDEA
 1. Choose Help | Obtain Permanent License in IntelliJ IDEA menu bar.
 2. When the Enter E-Mail Address dialog box opens, enter a valid e-mail address to which a temporary activation code will be sent.
 3. Click OK.
 4. When the Enter Activation Code dialog box displays, paste the activation code from the e-mail message that License Server sends to the e-mail address you specified. If the activation code is successfully validated, this will be confirmed by the Permanent Ticket Received dialog box.

When a client obtains a permanent ticket, its floating ticket is released.
When at least one permanent ticket has been issued, a table displays in the Permanent Tickets tab specifying e-mails to which activation codes were sent, as well as versions of client applications that received permanent tickets (fig. 9).

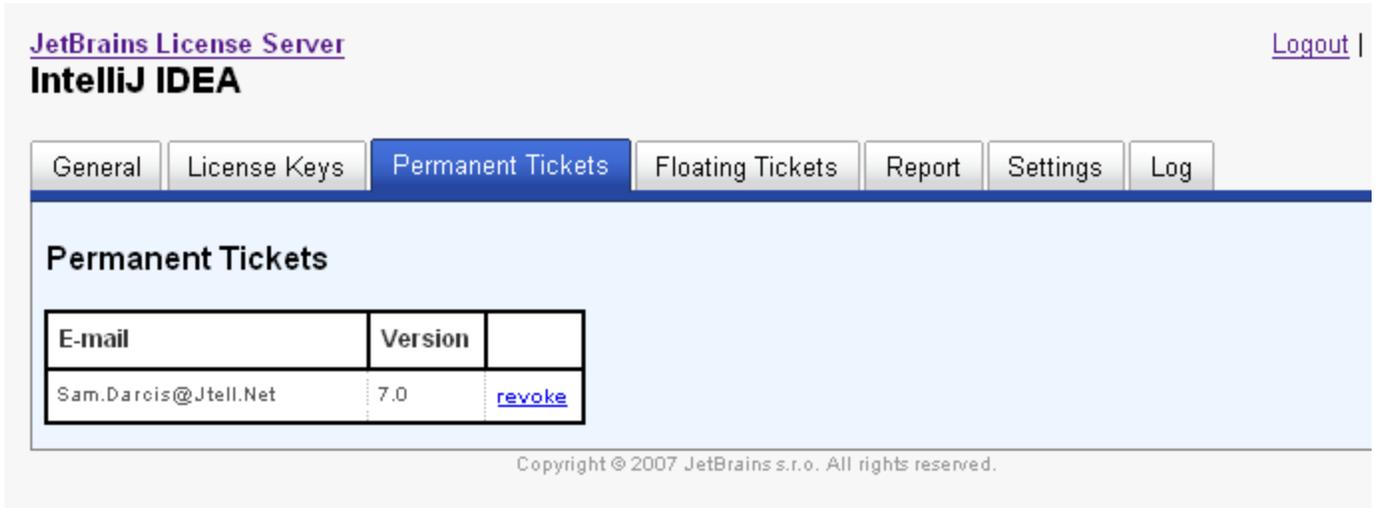


Figure 9. Product-Specific Control Panel | Permanent Tickets

To revoke a permanent ticket issued to a specific client, click the revoke link in the third table column next to this client's credentials. When the Remove Permanent Ticket? pop-up window opens, confirm or cancel the revocation.

Floating Tickets

This tab (fig. 10) contains a table with a list of clients that have received floating tickets, consisting of three columns:

- User Name: name of a user who is provided with a floating ticket.
- Licensee ID: a unique identifier of a product family installed on a specific client system.
- Version: product version.



Figure 10. Product-Specific Control Panel | Floating Tickets

- ✔ To obtain a floating ticket for a copy of IntelliJ IDEA
 1. Choose Help | Register in IntelliJ IDEA menu bar.
 2. Click Enter license server address and paste the URL or your license server in the corresponding text field.
 3. Click OK.

- ✔ To obtain a floating ticket for a copy of ReSharper
 1. Choose ReSharper | License Information in Visual Studio.
 2. Select Obtain from License Server.

3. Click Specify License server and paste the URL of your license server in the corresponding text field.
4. Click OK.

Report

This tab helps you generate reports on the usage of tickets within any time span. To specify the time span, use Start date and End date links. Click Generate to display a report table (fig. 11) with the following columns:

- Date: dates within the specified period when tickets were issued.
- <Version> (for example, 7.0): one or several columns indicating the number of tickets issued for a specific product version.
- Day Total: the total number of tickets issued for all product versions day by day.

JetBrains License Server [Logout](#)
IntelliJ IDEA

General License Keys Permanent Tickets Floating Tickets **Report** Settings Log

Tickets Usage Report

Start date: [23-09-2007](#) (d-m-y) End date: [23-10-2007](#) (d-m-y)

| Date | 7.0 | Day Total |
|------------|-----|-----------|
| 23-10-2007 | 2 | 2 |
| Total Max: | 2 | 2 |

[Full XML Report](#) [Short XML Report](#)

Copyright © 2007 JetBrains s.r.o. All rights reserved.

Figure 11. Product-Specific Control Panel | Report

The Total Max row displays the maximum daily number of tickets issued to individual versions as well as to all versions of the product within the specified period of time.

Full XML Report and Short XML Report links in the bottom part of the tab create signed XML files with reports previously generated in this tab. A short XML report contains product version names and the number of tickets issued to each of them. A full XML report adds the total number of tickets issued every day.

Settings

Two user controls are available in this tab (fig.12):

- Ticket Revocation Period (s): every client with a floating ticket must send a prolongation message to the server after a certain period of time has passed. You can specify that period in seconds in this field. If the client does not prolong its ticket within the revocation period (e.g. due to power outage or unexpected crash), the ticket will be revoked by the server and added to the pool of free tickets. The client that held the ticket, will have to re-obtain the new ticket. If the server is offline at the time the client attempts to prolong the ticket, it is up to the client how to handle this situation. All current clients usually tolerate server absence for some time (48 hours). If server is still offline after 48 hours, the client will ask for a license again.
- Enable Permanent Tickets: select this check box to allow receiving requests for permanent tickets from clients. Keep in mind that the Permanent Tickets tab remains hidden unless you select this check box.

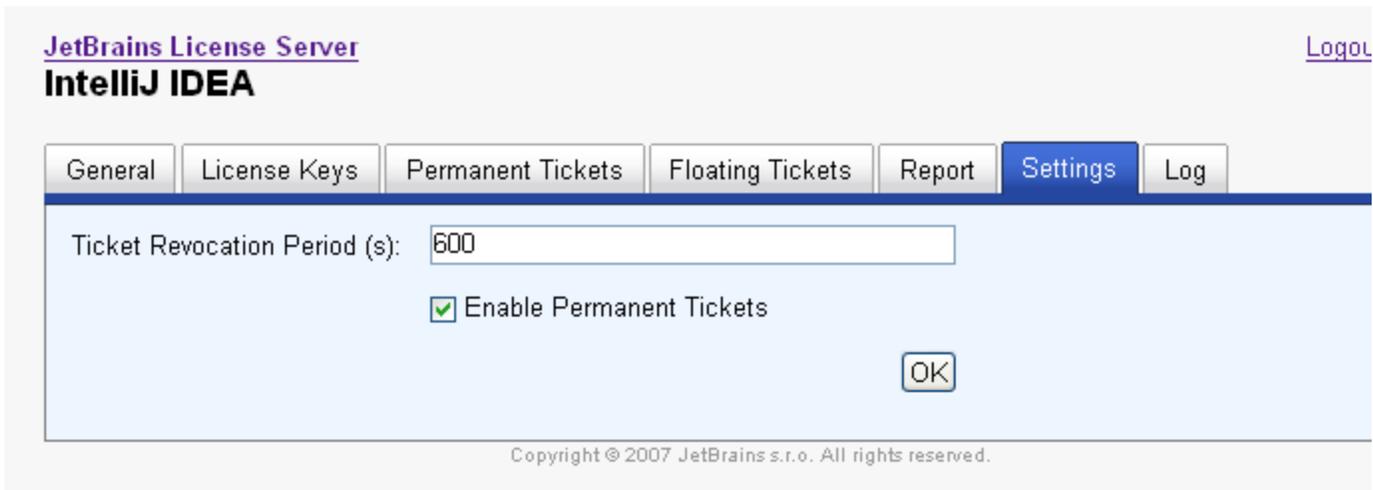


Figure 12. Product-Specific Control Panel | Settings

Clicking OK saves your settings.

Log

This tab (fig. 13) displays the contents of the log file maintained by License Server. The log file includes all ticket issue/release events along with additional data. Log records are generated with the following format:

Date Time [event severity] Action UserID, ticketID

For example:

2007-10-09 16:06:16,274 [INFO] Prolonged ticket for User.Name@unit-059.Int.YourCompany.Com, ticketId=5

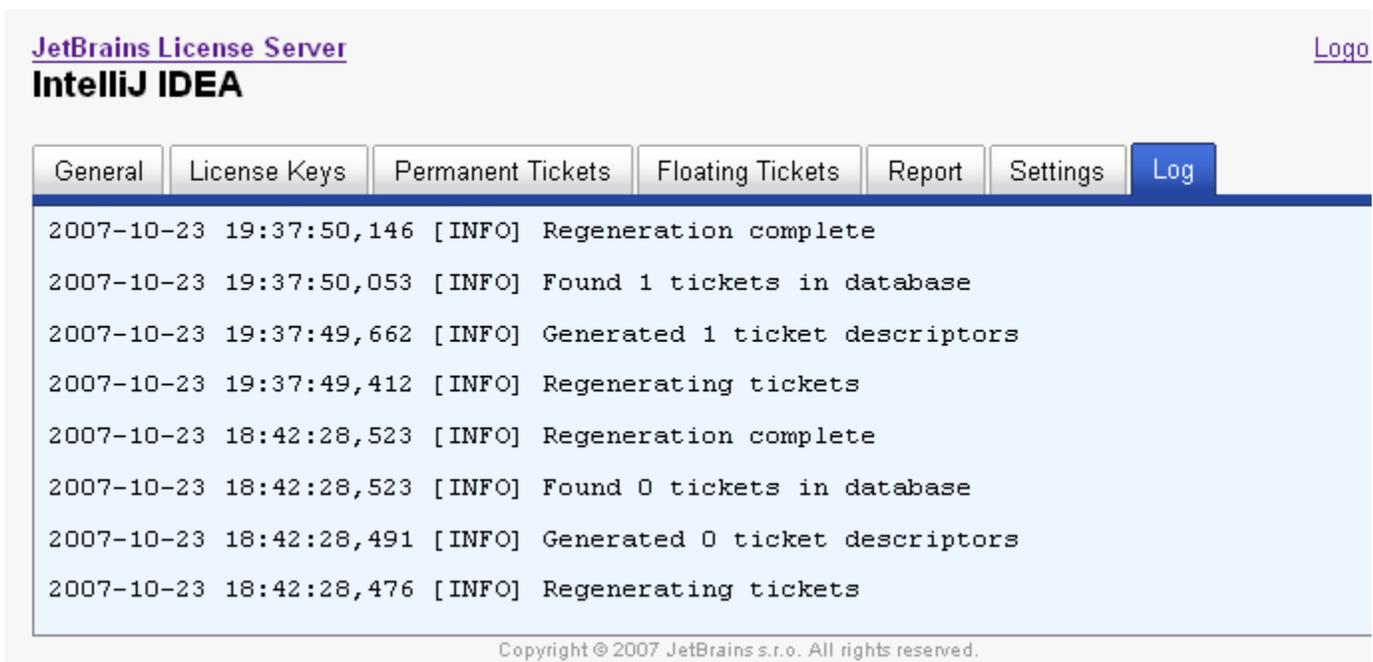


Figure 13. Product-Specific Control Panel | Log

Note that only 10 Kb of the log's most recent records are displayed in this tab. You can view the entire log by opening it from the Tomcat root directory. Note that for every product managed by License Server, a separate log is maintained under the name <ProductID>.log Where ProductID is the unique product family identifier that can also be found in the URL of the corresponding product-specific control panel in License Server.

You can generate custom reports that match your specific needs by retrieving specific data from the log files using awk scripts.

You can learn more about awk scripts at <http://www.esmerel.com/wagons/rob/awkwords.html>.