

Eclipse Passage Floating License Server Administration Guide

July 18, 2021



ArSysOp

Contents

| | |
|--------------------------------------------|----------|
| Overview | 1 |
| Contribution | 2 |
| License | 2 |
| How floating licensing works | 2 |
| Floating License Server | 3 |
| Communication with the Server | 3 |
| Full floating licensing workflow | 3 |
| What is in a Floating License Pack | 4 |
| Pack | 5 |
| Files | 5 |
| Content | 5 |
| Floating license file | 5 |
| Floating access file | 7 |
| Destination | 7 |
| Install Floating License Server | 7 |
| Run Floating License Server | 8 |
| Command line parameters | 9 |
| Commands | 9 |
| Lifecycle commands | 9 |
| Self licensing commands | 9 |
| Manage Floating License Packs | 10 |

Overview

Eclipse Passage is a set of tools and libraries providing rich and easily adaptable capabilities to declare and control licensing constraints.

Contribution

If you found a mistake in the text or just want to improve this user guide, feel free to contribute on [Github](#) repository.

Also we want to remind that Eclipse Passage is an open-source project, so you can always contribute to the project [itself](#).

License

These materials are made available under the terms of the [Eclipse Public License 2.0](#).

How floating licensing works

As opposed to personal licensing, where a user of a product-under-licensing gets their own license and hosts it close to the product for single use, floating licensing delegates actual license packs hosting and executing to a dedicated agent - **Floating License Server**.

This allows number of users to exploit their installations of the product simultaneously in the scope of a single floating license pack.

Floating License Server

The Server, that is installed somewhere in the reach,

- hosts all floating license packs and
- responds to *give me permission to use this feature* requests from the product's users via http.

Communication with the Server

The product

- asks the Server for a grant on a particular feature and,
- if succeed, proceeds with the feature and then,
- when it's over, releases the grant.

The Server can lease several grants for the same feature at the same time, as hosted floating licence packs permit.

Full floating licensing workflow

There are four *actors* in the floating licensing workflow:

- *product development and management*, where
 - management *defines products, feature sets* and *product release plan* and
 - development
 - * *declares licensing requirements* in the product codebase,
 - * *covers feature usages with licensing protection*,
 - * *configures the product to use Floating License Server*;
- *operator*, who issues a floating license pack and delivers it,
- *agent*, who installs and runs a Floating License Server,
- *client* as a representative for a set of physical users of the product *users*, which are enlisted in a *floating license pack*.

What is in a Floating License Pack

Pack

When operator issues a floating license pack, it resides in a folder named after the *pack identifier* like *d2b83215-b65d-4031-a8c8-a10421d56260*. Name of each file in the folder starts with this identifier as well. The folder contains all necessary data for all parties involved in floating licensing.

Files

There are in the folder

- **license files**, which contain list of all users eligible for the license and any feature can be used by it; they should be passed under control of **Floating License Server**
- optionally, **floating access files**, one per user, which keep connection and authentication credentials of the Server; there files are to be passed to each user mentioned in floating the license pack.

Table 1 shows a typical file structure of a floating license pack, delivered to a client.

Content

Floating license file

Floating license File **.flicen* accumulates

Table 1: Sample floating license pack content

| File | Role | Action |
|---------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------|
| <i>pack-id.flicen</i> | Encrypted license | Pass to agent to be deployed under Floating License Server. |
| <i>product-id_product-version.pub</i> | Product's public key | Pass to agent to be deployed under Floating License Server, together with the license file. |
| <i>pack-id_user-id flaen</i> | Encrypted floating access file | Pass to the mentioned user to be stored close to their installation of the product. |

- **Floating License Server authentication** information, as each floating license is to be released for a particular residence. A license is not intended to be functional in case it is deployed on a server, that cannot be authenticated with these credentials;
- enlistment of all **users allowed to exploit the product** - their identifiers, authentication conditions, other info;
- list of product features that can be used in the scope of this license pack, associated with the amount of available grants (simultaneous usages).

A product, running on a *user's* installation, *acquires a grant* for a *feature*, which is valid for short period (*vivid* for 60 minutes by default). Since a grant is acquired, it's not accessible until it is released.

After the *feature's* work is over, the product *releases the grant* and thus it becomes available for acquisition again.

Feature grant capacity

Several product installations (users) can acquire different grants for the same feature for overlapping time slots. Feature grant's *capacity* field defines this amount. If there are no more available grants, the acquisition fails, and the product blocks the feature utilization.

Feature grant vivid

A feature grant, been leased for a user, should not last long. Thus, a feature grant information contain instruction on how *vivid* is the grant. Measured in minutes, it defines how long will the grant be valid after leased (limited by the whole license pack validity period). After this relatively short vivid period the grant is not active, but still acquired until is explicitly released.

Feature grant validity period

A feature grant can define its own validity period. Even if it is declared to be wider than the owning license pack validity period, at runtime the former is still limited by the latter. This feature is used by licensing party to narrow validity period for a particular feature in the pack.

Floating access file

A product installation, been located on a user's workstation, needs to be told how to access a Floating License Server.

Floating access file delivers this information. It contains:

- user identifier,
- the Server connection url components: ip address and port,
- the Server authentication condition.

Each user enlisted in the license file should possess such an access file and host it where the product is configured to look for it. Usually it is the product's folder under default *.passage* data directory: *user-home/.passage/product-id/product-version*

Destination

Floating license files are to be put under control of the Floating License Server for which this floating license pack has been issued.

Install Floating License Server

Floating License Server is distributed in OS-specific archives, which does not need to be installed, just unpacked.

Nevertheless, server executable should be granted with all the necessary permissions

- to its working directory,
- to the directory where the Server hosts license packs.

The Server is implemented over **Java 11**, so make sure that *java -version* reports proper version and this runtime is available for the server executable.

Run Floating License Server

Start the Server executable located in the root of the unpacked content. It triggers *osgi* console and there immediately starts the Server. The console is used for the Server administration.

Command line parameters

Server runtime can be configured with the command line parameters for the Server executable, enlisted in Table 2:

Table 2: Floating License Server command line parameters

| Parameter | Default | Usage |
|------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------|
| <i>-server.port</i> | 8090 | <i>-server.port=8089</i> will start the Server which will use the specified port for http communications. |
| <i>-server.storage</i> | <i>user-home/.passage</i> | Path to directory where the Server is expected to keep floating license packs. |

Commands

Lifecycle commands

The Server can be stopped and restarted from the console.

Table 3: Server Administration Commands | Lifecycle

| Command | Output |
|----------------------|-----------------------------------------------------------|
| > <i>fls:stop</i> | Stops the Server |
| > <i>fls:start</i> | Starts the Server |
| > <i>fls:restart</i> | Restarts the Server |
| > <i>fls:state</i> | Reports either the Server is currently started or stopped |
| > <i>exit</i> | Terminates the Server and exit the administration console |

Self licensing commands

Floating License Server itself is a product licensed with Passage. That means you should put proper personal license for the Server under *user-home/.passage/server-product-id/server-product-version* folder.

Table 4: Server Administration Commands | Self Licensing

| Command | Output |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------|
| > <i>self:licstatus</i> | Reports current licensing status of the Floating License Server product. |
| > <i>self:licrequest</i> | Reports complete platform assessment, which is necessary to request a license for the Server from a licensing operator. |

Manage Floating License Packs

Upload Floating License Pack

Use *fls:upload* command to put a floating license pack under the Server control.

```
> fls:upload path/to/a/fl-pack-folder/like/0803cf43-b9ce-4051-bcb8-15ccf00f036b/
```

The Server reads the pack and places **.flicen* and **.pub* files under the directory, condifured by *storage* parameter.