Eclipse Passage Operator User Guide

July 6, 2021

ArSysOp

Contents

Introduction **1**
- Overview . . . 2
- Contribution . . . 2
- License . . . 2

Components **2**
- Licensing Components . . . 3
- Operator . . . 3
- Floating License Server . . . 3

Developer's guide **3**

Eclipse Operator User Guide **4**
- Introduction . . . 5
  - Main concepts . . . 5
  - UI Overview . . . 5
- How to license your product . . . 6
  - Definition . . . 6
  - Evolution . . . 10
  - Licensing . . . 10

Glossary **10**

Introduction

Overview

Eclipse Passage is a set of tools and libraries providing rich and easily adaptable capabilities to declare and control licensing constraints. Being an open-source project under Eclipse Foundation, it is entirely developed with Java 11 by the *ArSysOp* company.

This user guide contains a glossary, which gives you base understanding of terms used in Eclipse Passage and two sets of instructions: for developer, who wants to license some features or entire application with Passage, and for operator, who wants to manage licenses and metadata with the Eclipse Passage Operator.

Happy Reading!

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

Components

Eclipse Passage consists of three parts:

Licensing Components

Licensing Components is a set of tools you have to include in your application to be able to declare any licensing logic (For example, what to do if a correct license was not acquired by the licensing framework). It allows you to restrict unauthorized using of a single feature, bundle or even the whole application.

Operator

Operator client is a separate application giving you a manual control under all the licenses your product can have (both personal and floating). Also, it is required in access cycle in order to define the product or/and features metadata and to generate keypair for license file encoding.

Floating License Server

Eclipse Passage FLS is a component implementing the Floating License Server concept. Works roughly in a way described below.

The licensee acquires (in any way) a finite number of licenses and these licenses are stored on the License Server. When an authorized user wants to use the application, they request a license from the server, and if the license pool is not empty (in other words, there are still licenses available), the server gives user an access to the application. When user finishes his work with the application, the license is released and can be obtained by the other authorized user.

Developer's guide

This section is under construction for now.

Eclipse Operator User Guide

Introduction

Main concepts

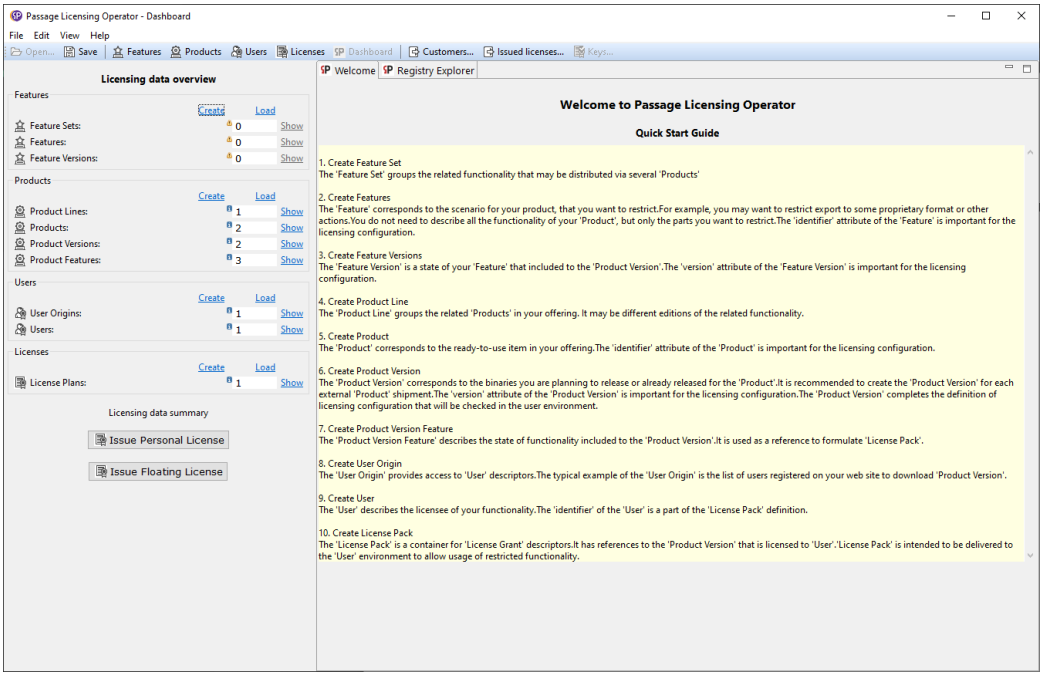
The best approach to learn any existing user interface is understanding the concepts it is based on. In Eclipse Passage worldview the product licensing sequence consits of three parts:

- **Definition** - on this step you define your product lines, products and all the features it has to be licensed with Eclipse Passage. Responsibility for this step usually lies on the marketers' shoulders.
- **Evolution** - when you defined the product and features it has, it's time to start development. New versions are being released, some functionality can be added, changed or even dropped away.
- **Licensing** - after you have at least one version of your product or feature released, you can start issuing licenses to your users.

These are three main steps to have your features properly licensed with Passage Operator.

UI Overview

Eclipse Passage Operator's initial layout is quite simple: dashboard with all products, features, users and license plans contained in your workspace; workbench part (initially filled with welcome page), where you you usually deal with some licensing data (E. g. you can define your features here), top toolbar with some useful wizards and controls, two buttons to issue a license.



How to license your product

This tutorial contains three parts mapped with three licensing steps mentioned in the section above.

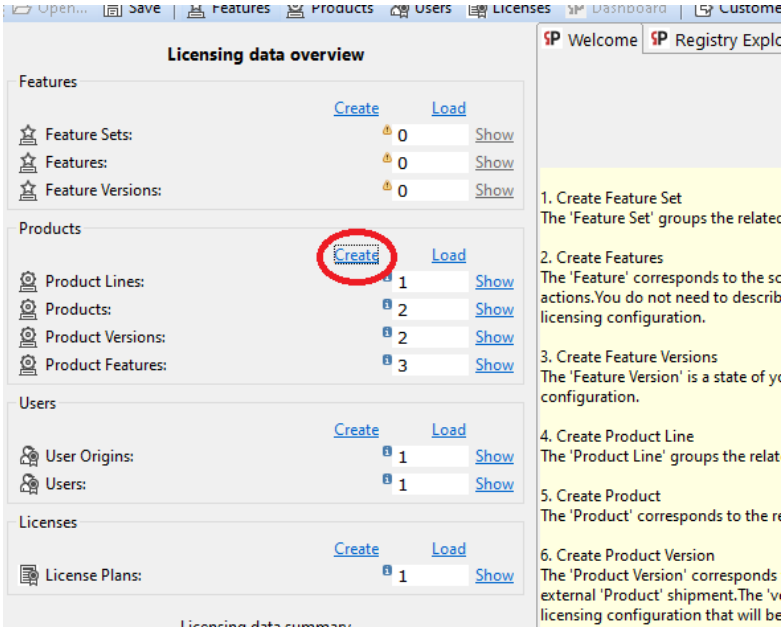
Definition

First of all, we have to define what our product is. To be more concise with the reader, tutorial at all is provided with examples based on Eclipse Passage commercial version (Cordon).

Usually some products are provided separately around the same system (library or something like that). In our case ArSysOp Cordon provides a simple product line, containing three licensable components:

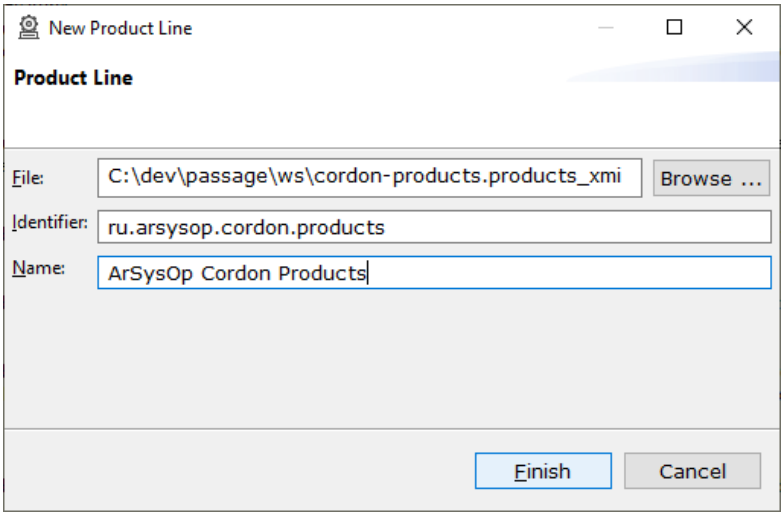
- Cordon Access Cycle (Licensing Components)
- Cordon Operator (Like Passage Operator)
- Cordon Floating License Server

Let's open product line creation dialog by pressing the *Create* button near the Products area.



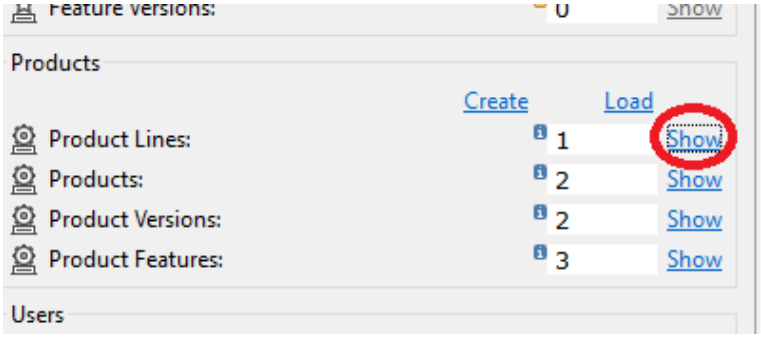
In the dialog appeared we have to choose a place, where our product line model will physically exist (as .xmi model), choose unique identifier and any beautiful name.

Remember location where your xmi files are saved. These model files can also be imported with *Load* button.

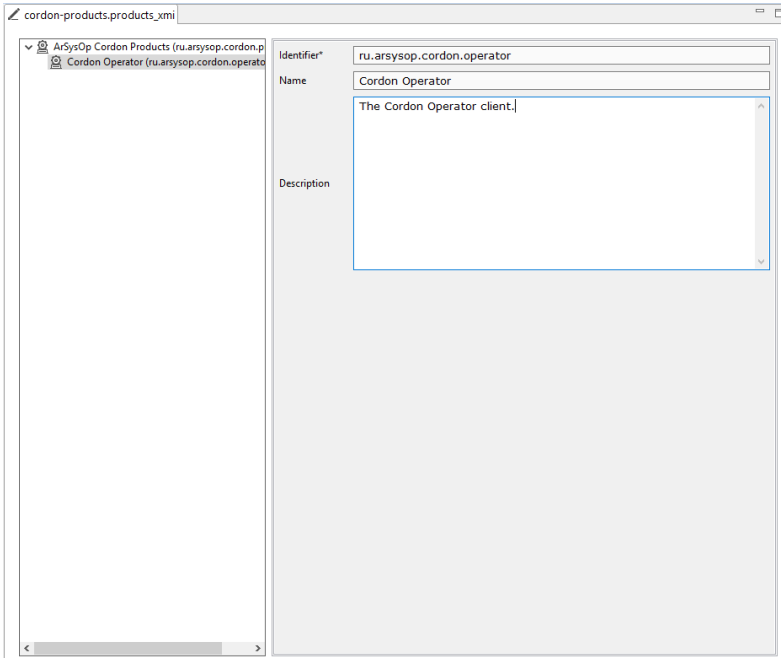


Press *Finish* and that's it! My congratulations, you have just defined your first product line with Passage! Let's fill it with some products.

Press *Show* near the Product lines count. Now you see the workbench with one defined product line and no products under it.



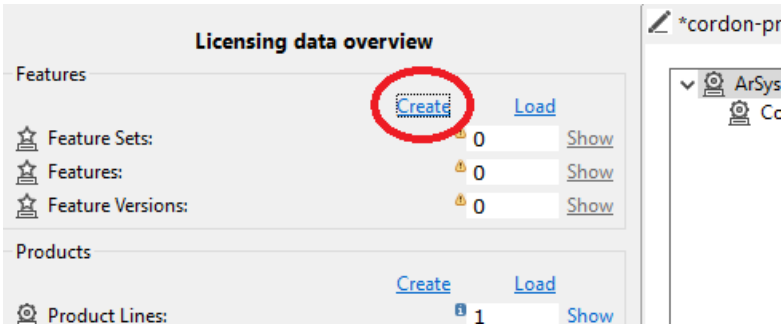
On your product line with right click open the context menu, where you will see an option named *Product*. Pressing it you define new product under the selected product line. Let it be the Cordon Operator in our case. Fill all the required fields and optional fields if needed.



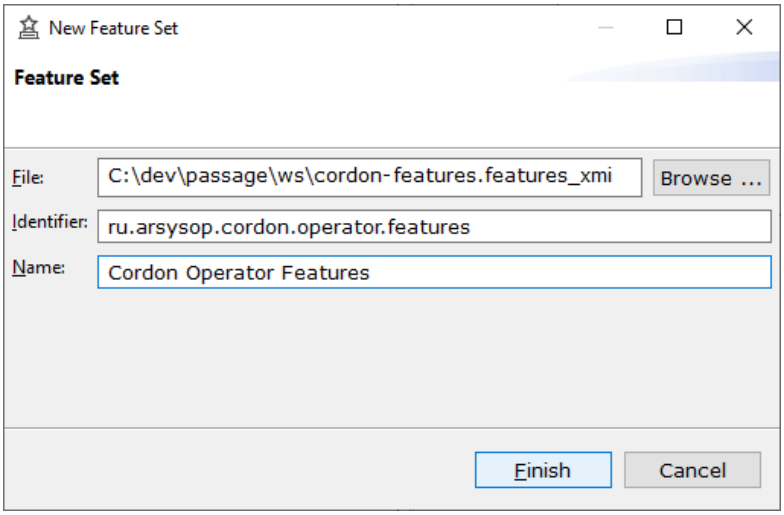
Congratulations again! The product is now created and ready to be filled with features, which we are about to define. Let's imagine we have two licensing cases for our product:

- we want to check license on start for all users
- we want one feature to be licensed separately - not all users who bought Cordon Operator License are allowed to use it. Let it be license usage statistics, for example.

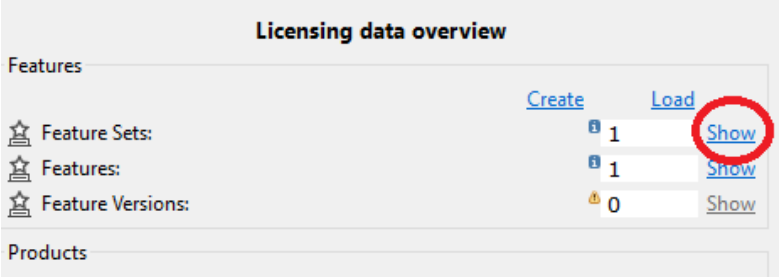
First, we have to define a feature that will represent usage statistics collection. Press *Create* button near the Features area to create Feature Set.



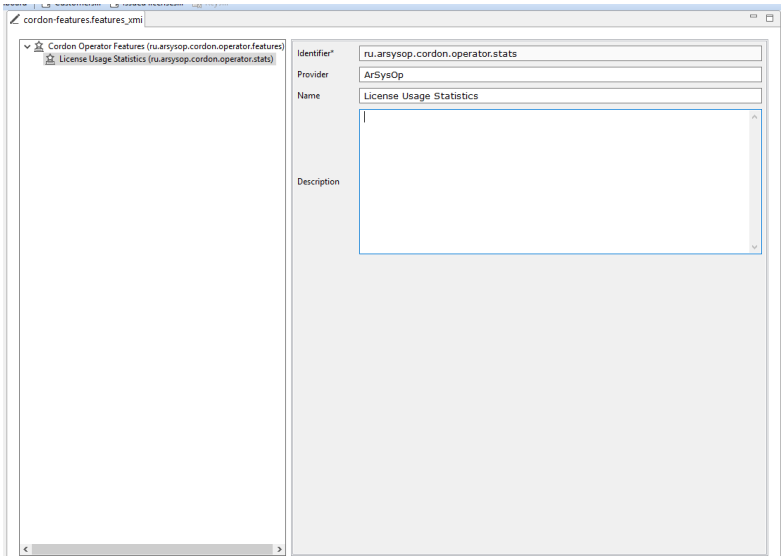
And then select model file location, feature set id and name.



Press *Finish* and then *Show* near the feature set count.



Like the product under product line, create the feature under feature set with the context menu showed on right-click, then fill identifier and name fields at least and save.



Now we successfully defined our product exceptional feature. To have our product licensed at all, we need to define another feature with **the same identifier as the product has**. Also, developer must use the same identifier when they define their product with Eclipse.

For now we have one product line, one product in it and some exceptional feature to be licensed separately. That's quite enough to proceed to the next part - **Evolution**.

Evolution

Licensing

Glossary

Feature

Feature Set

Feature Version

User

User Origin

License Plan

Product

Product Feature

Product Line

Product Version