[redacted]

exactpro
EXITUS ACTA PROBAT

CONFIDENTIAL

# JackFish

General Information

| | |
|---|---|
| Version | 1.0 |
| Release date | 09 Jul 2018 |
| Pages | 6 |

# Contents

# 1 Document Information

## 1.1 Abbreviations

| Abbreviation | Meaning |
|---|---|
| JF | JackFish |

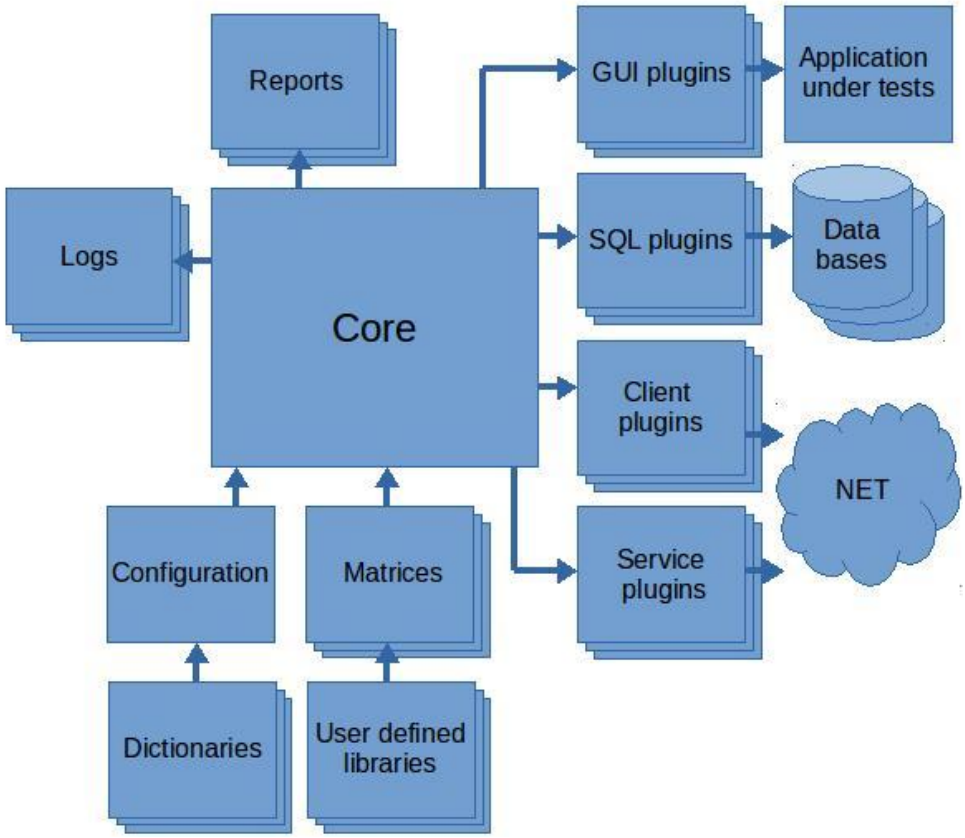
# 2 General information

## 2.1 Introduction

JackFish ("JF" further in this document) is a tool for test automation of web-based applications built on Selenium framework. JF can perform positive and negative functional testing, GUI testing, verification of a variety of data formats (xml, csv, text, etc.). JF produces a detailed report about performing each test case. It allows QA engineers without software development experience to automate test scenarios. In order to achieve that, JF uses a simple csv format expression language where rows contain actions that manipulate GUI controls on web pages, and the columns contain parameters for each such action. Additionally, to simplify the process, JF introduces a dictionary entity that describes all GUI controls and methods for locating them on a web-page, and also assigns them names that are referred to in test cases. The solution also includes GUI editors for preparing dictionaries and test scenario files. The tool supports a GUI-less mode of running it and can work with multiple browsers simultaneously.

## 2.2 Architecture

The following diagram explains the JF architecture on a high level:



## 2.3 Test Cases Preparation

The file with test scenarios has the following csv format:

```
#TestCase
Login

#Id;#Action;#AppId;#Browser;#URL
APPSTR1;ApplicationStart;'EC';'Firefox';'https://some.site.com'

#Global;#Action;#AppCon;#User;#Password
1;Vars;APPSTR1.Out;user;password

#Id;#Action;#Dialog;#AppConnection;#Email;#Pass;#SIGNIN
DLGFLL2;DialogFill;'BeforeLogin';AppCon;User;Password;
```

Where: `#AppId, #Browser, #URL` and etc. is the header of an action that describes the names of parameter; `ApplicationStart, Vars` and etc. is the actual action that manipulates a GUI control or verifies some parameters retrieved from a web page.
Each cell in csv in the file can contain an MVEL language expression. This allows calculating math expression and calling any Java language function. This gives the possibility of creating very complex and flexible scenarios. There is also a possibility of using `#If` and `#For` statements as well as sub routines that can described in a separate file.
It is possible to edit scenarios in any text editor, but we find that built-in JF script editor is the most comfortable. The editor can check the correctness of all expressions before running the matrix. It also helps users to fill required values faster.
A GUI dictionary is an xml file in that describes all dialogs (web pages) and GUI controls the users need. In test scenarios, the users refer to GUI controls from dictionaries using only their names. The dictionary allows separating the process of searching GUI elements from the business logic. The dictionary can be filled manually or by capturing the elements automatically from the application under tests.

## 2.4 System Requirements

The configuration of JF is in a quite simple xml file where all needed entities are described. It includes plug-ins, dictionaries, routine user libraries, etc.
*The systems requirements are as follows:*
Linux or Windows with Java SE 1.8
at least 1 GB RAM for comfortable work

10 GB HDD for the tool and log files

## 2.5 JF Improvements Pipeline

The JF software development team is working on new features to make work of our users more comfortable. In particular, the following is going to be implemented:

1. Support of responsiveness testing (the same functionality as Galen framework);
2. Support of GUI testing for Windows applications with UI Automation technology.