

Table of Contents

Table of Contents	1
ISSUE-TRACKER SYSTEM GUIDE	2
1. INTRO	2
1.1. Purpose	2
1.1.1. Audience	2
2. ARCHITECTURE	2
2.1. IOCM architecture definition	2
2.1.1. The Control components	2
2.1.2. The Model components	2
2.1.3. The Input Components	2
2.1.4. The Output Components	2
2.1.5. The Converter Components	2

# ISSUE-TRACKER SYSTEM GUIDE

## 1. INTRO

### 1.1. Purpose

The purpose of this guide is to provide description of the existing issue-tracker System and it's architecture

#### 1.1.1.

##### Audience

Any given instance of the issue-tracker should have ONE and only ONE person which is responsible at the end for the functioning of THIS instance - so think carefully before attempting to take ownership for an instance. The author(s) of the code are not responsible for the operation, bugs or whatever happens to a new instance. As a responsible owner of an instance you could create, share and assign issues to the authors of the source code, yet there is no service level agreement, nor even promise to help.

## 2. ARCHITECTURE

### 2.1. IOCM architecture definition

The Input-Output Control Model architecture is an application architecture providing the highest possible abstraction for almost any software artifact, by dividing its components based on their abstract responsibilities, such as Input, Output, Control and Model.

#### 2.1.1. The Control components

The Control components control the control flow in the application. They instantiate the Models and pass them to the Readers, Converters and Writers for output.

#### 2.1.2. The Model components

The model components model the DATA of the application - that is no configuration, nor control flow nor anything else should be contained within the model.

Should you encounter data, which is not modelled yet, you should expand the Model and NOT provide different data storage and passing techniques elsewhere in the code base ...

#### 2.1.3. The Input Components

The Input Components are generally named as "Readers". Their responsibility is to read the application data into Model(s).

#### 2.1.4. The Output Components

The Output Components are generally named as "Writers". Their responsibility is to write the already processed data from the Models into the output media.

#### 2.1.5. The Converter Components

The Converters apply usually the business logic for converting the input data from the Models into the app specific data back to the Models.