

# Log Analyzes 说明文档

## 01. 功能概述

用来读取软件中的日记信息，用来整理和提取日记中的信息，并输出到指定位置，方便对日记进行分析

## 02. 使用方法

# 将日记内容输出到 MySQL 数据库

```
bin\IDOL-Query.py -f logfile -t tablename [-d dbname | -P 3306 | ..... ]
```

# 将日记内容输出到 csv 文件

```
bin\CBK-Agent-Summary.py -f logfile -out csv
```

# 将日记内容分析并生成分析结果

```
bin\MSSQL-ErrorLog.py -f logfile -out report [-detail on]
```

## 03. 定制方法

### 编写启动程序：

# 参照 **doc/example\_bin.txt** 模板添加 bin 目录下的启动程序，主要修改 **main** 函数的 p2 和 p3 里面的内容

# p2 是日记分析模块

# p3 是输出端方法

### 编写分析模块：

# 参照 **doc/example\_mod\_analysis.txt** 添加日记分析模块

# 模块定义的时候至少需要定义 2 个变量，分别是 2 个消息队列 (queue)

# 第一个 queue 是从启动程序获取日记信息

# 第二个 queue 是将分析的结果放到队列中，并且由下一个进程来处理

# 如果想要输出端为 Report 模式，请参考“**根据输出端编写其余代码**”其中的内容

# 可以参考 **mod/analysis/General** 模块中的写法，该模块会加载一个规则文件

# 如果 General 里面的这个方法符合您的需求，则可以直接编写规则文件即可

# 如果不符合您的需求，则可以自定义一个日记分析模块，可以参考如下模块：

# **mod/analysis/SQLDB** 这个模块，输出端适用于 Report

# **mod/analysis/IDOL 或 ConnectedBackup** 这个模块，输出端适用于 MySQL, CSV

### 根据输出端编写其余代码：

#### To MySQL

# 由于输出到 MySQL 需要创建数据表，所以需要在 **mod.tools.TemplateMySQL** 类中添加一个创建表的模板

# 接收字典类型的数据

#### To CSV

# 由于输出到 CSV 时需要创建 headers 信息，所以需要在 **mod.tools.TemplateCSV** 类中添加一个 headers 的模板

# 接收字典类型的数据

#### To Report

# 需要编写规则文件，规则文件是个列表，里面都是字典数据

# 可选字段：name，该字段暂是为了方便阅读性，可以不添加

# 每个字典中需要有如下 key 的信息：

# type: 值可以为 Information、CPU、Memory、Disk、Network、Permission、EventID、Others

**当 type 的值是 Information 或 Others 时，每个字典必须包含如下字段：**

# info: 这个问题的标题

# keyword: 匹配这个问题的关键字（正则表达式）

# rule: 如果匹配该行，则值是什么？其中 line 代表该行日记，类型是字符串，例如：line.split('=')[1]

当 **type** 的值是其它的值时，即默认情况下，每个字典必须包含如下字段：

# info: 问题原因

# keyword: 作为匹配此条规则的关键字（正则表达式）

# solution: 解决思路

#### 04. 授权模式

GNU General Public License v3.0