Log Analyzer 1.0

1. Get Source code from: [\\yanju03-work\LogAnalyzer\LogAnalyzer\](file:///\\yanju03-work\LogAnalyzer\LogAnalyzer\)

2. Get installation from: [\\yanju03-work\LogAnalyzer\LogAnalyzer\Installation\](file:///\\yanju03-work\LogAnalyzer\LogAnalyzer\Installation\) Unzip the file LogAnalyzer.zip to any folder, then run the LogAnalyzer.exe or LogAnalyzer\_Excel.exe directly (It can generate report in excel file)

2. Requirements:

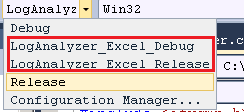
a. Install Python, and its version at least 3.3 or later. If build 32bit application, you should install the 32bit python

b. Install MS office 2010 or higher version, but this is optional.

c. Notepad++. We need use this tool to open source file. We will support more the 3rd tools.

3. Build

Please see below screenshot, there are four build configurations, the two labeled with red color contain the functionality of generating report in excel file, so you have to install MS office in your local build.

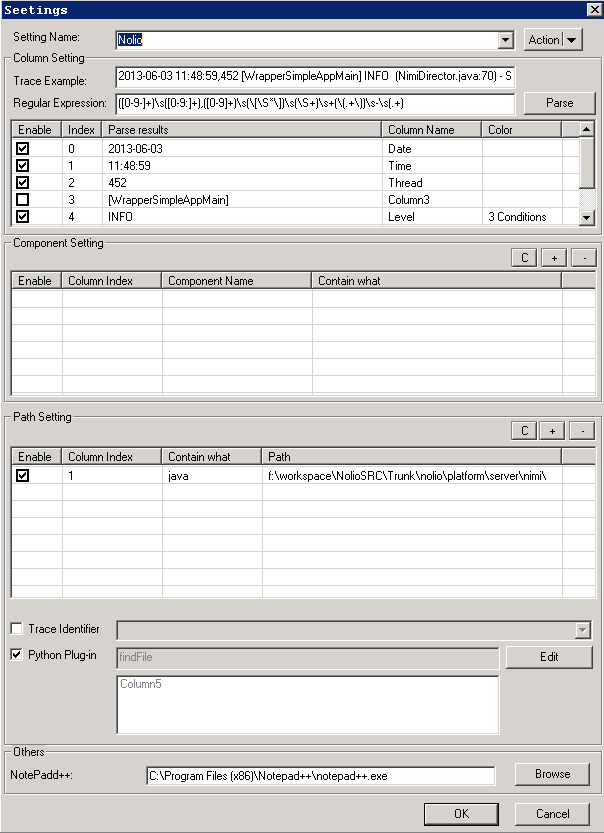


4. Introduction

Generally, there is a folder ‘Setting’ in the same directory with logAnalyzer.exe, it contains all configuration files you defined. When the application starts, all configuration files will be loaded. You also can edit/create configuration in UI.

5. Setting

Click the ‘Settings’ to open the dialog to edit configurations



How to create a new configuration?

a. Click menu action, select ‘New’.

b. Input a log example, Input a regular exception to parse the log example, click the ‘Parse’ button. The regular exception is used to divide the log example into several parts. We call the each part as column.

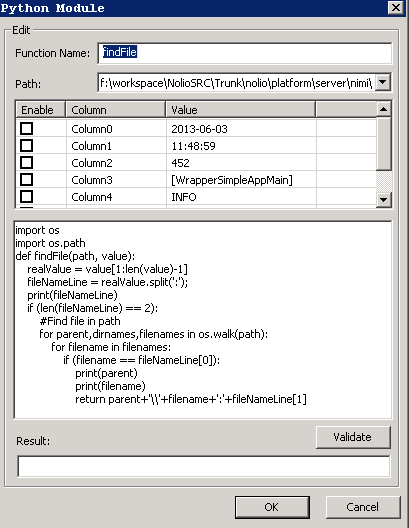
c. You can enable/disable the columns show in the list control. If a column is disabled, it will not show in the main windows. You also can rename the column name, double click the ‘Column name’ to edit. In order to show different color for column, you can add conditions to any column, double click the ‘Color’ item to edit.

d. Component setting. The component is used to manage the log easily. Click ‘+’ / ’-’ to add/delete. The component uses the key word to identify, the key word is case sensitive.

e. Path setting. This part is used for locating log in source code file.

First, please set a path, the path is a range.

Second, specify the log identifier. You can get the identifier in the log content, for example, the rainer.log, each log has a unique trace id, so the trace id can be the identifier. But for some log files, you cannot find identifier from log directly, so for this scenario, you can write a python module to help get the file full path, you can choose ‘Nolio’ Configuration as an example to check more.



f. Notepad++.

Specify the path of Notepad++’s EXE file. We will use this tool to open source code and log file.

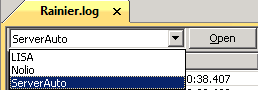
6. UI

If you want to analyze a log file, you can follow below steps:

a. Create a new document. A new tab is created.



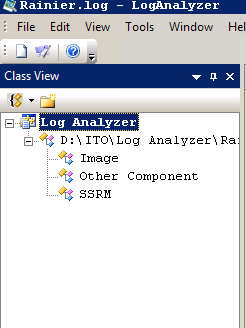
b. In the new tab, select what configurations you want to use.



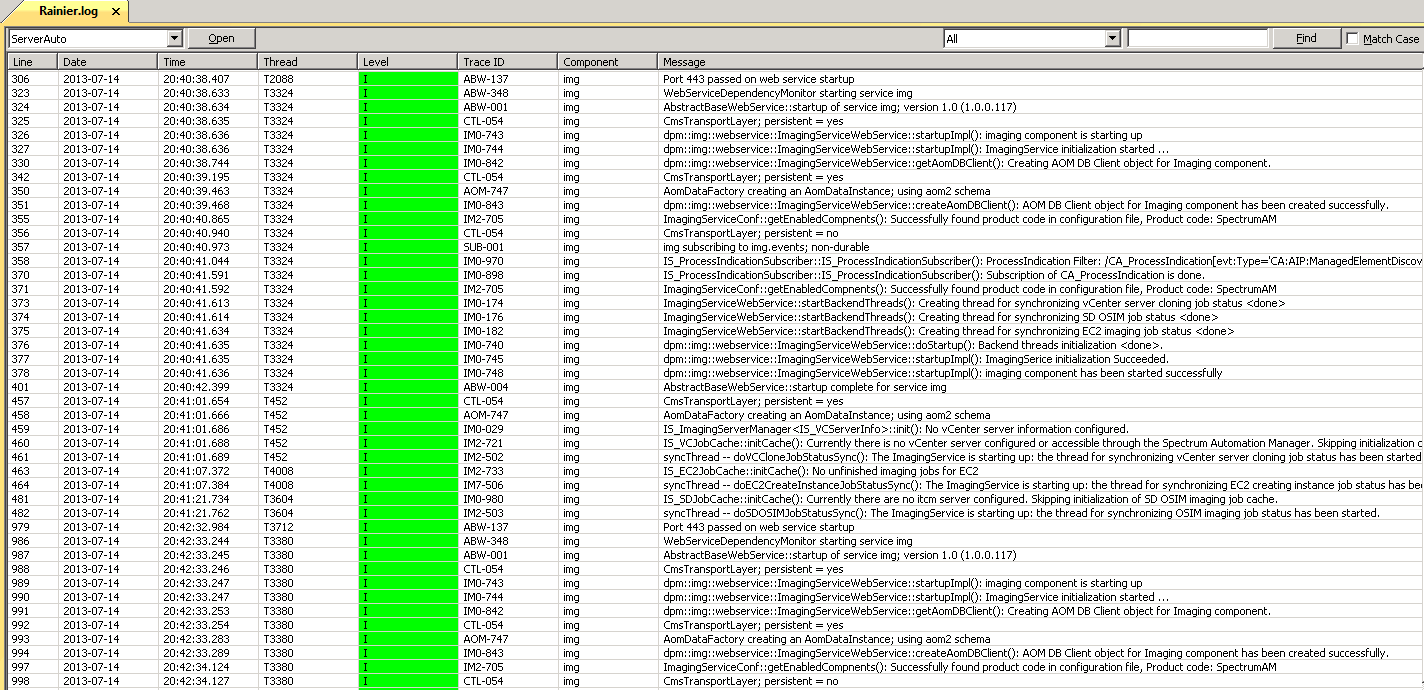
c. Click Open button, select what log file you want to analyze, then analyzing



d. Show opened log files and components

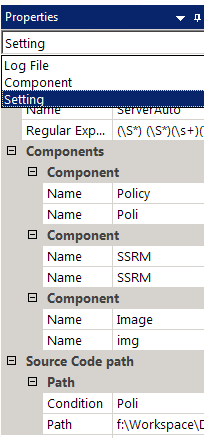


Click the component, then the list control will show all logs belong to the selected component.



e. Property

The right part is the property panel.



It has three panels for Log file, Component and the current Setting. The Setting panel shows the properties about the current setting used by the active opened log. They are read only, but it will support editing in real time in the feature.

f. Output

The output has three tabs, this part is implementing.

Summary : the basic information after analyzing a log file

Warning : All warnings during operation.

History : The history of operation