Performance Logger

The Performance Logger is a useful debugging tool for logging the job's performance statistics which are generated when you activate the Performance Reports. It writes jobs to a log file (if a log file is specified), generates an average of execution paths detected and also captures the last "numJobs" jobs in memory to be displayed on the admin interface. This component an extension of Job Logger.

The job is logged after it has completed its execution, this stage register itself as a job listener. This is done to log all the performance reports generated in all subsequent stages after this one.

To enable/disable this feature visit Performance Reports.

WARNING: This job logger creates all the directories required to create the logFile. Naturally, this makes it easier to use, but it does mean you need to be careful about where the log file is stored.

Performance Logger	
Factory Name	com.searchtechnologies.aspire:aspire-tools
subType	performanceLogger
Inputs	The entire jo.
Outputs	The input job's performance statistics are sent, unmodified, to the output, after loggin.

Where to use this component

This is an Aspire Stage, so it must be added to the execution path of the jobs in interest. Anywhere in that execution path would work, but we recommend using it as the first stage executed for consistency among all applications.

Configuration

Element	Type	Default	Description
logFile	String	null	(Optional) Specifies the file name to which jobs are logged. If missing, then no log file will be created. Note that directories required for the log file will be automatically created. Automatically creates the directories necessary to contain the log file if they don't already exist. If the log file already exists, first it checks for a ".bak" version of the file. If that exists it is deleted. Then it renames the current file to ".bak", and creates the new file.
enable	boolean	true	If true, enables logging and average time calculations.
numJobs	int	5	Specifies how many of the most recent jobs will be displayed from the admin interface. Set this number to zero if you don't want to keep any files in memory (i.e. just write to the log file), which you may prefer to do if your documents are very large and therefore may use up a lot of memory.

Example Configuration

Simple

```
<component name="PerformanceLogger" subType="performanceLogger" factoryName="aspire-tools">
  <enable>true</enable>
</component>
```

Complex

```
<component name="PerformanceLogger" subType="performanceLogger" factoryName="aspire-tools">
  <enable>${debug}</enable>
  <logFile>log/${app.name}/performanceStatistics.jobs</logFile>
  <numJobs>10</numJobs>
</component>
```

Example Output

```
...
```

Downloading of Performance Reports

To download the reports generated by this component you need to go to the component's debug console page.

Once in the debug console page you can select the format of your choice to download the report, and then click on the button named **"downloadExecutionPaths"**.

blocked URL

An execution path is the ordered list of applications, pipelines, stages and workflow rules in which each document was processed. There can be different execution paths for documents scanned from the same connector, this is because there can be different processing rules for different types of documents (such as Folders/Files). You can also create new execution paths by adding some flow control rules to your workflow.

Example Output

In CSV format:

```
name,type,averageTime,jobCount
RootExecutionPath,executionPath,10927,1
/FeedOneExample/StandardPipeManager,pipelineManager,10926,
doc-process,pipeline,10926,
/FeedOneExample/StandardPipeManager/PerformanceLogger,stage,0,
/FeedOneExample/StandardPipeManager/FetchUrl,stage,5595,
/FeedOneExample/StandardPipeManager/ExtractText,stage,5330,
/FeedOneExample/StandardPipeManager/ExtractDomain,stage,0,
/FeedOneExample/StandardPipeManager/PrintToFile,stage,0,
```

In XML format:

```
<executionPaths>
<executionPath averageTime="10927" count="1">
  <pipelineManager averageTime="10926" name="/FeedOneExample/StandardPipeManager">
    <pipeline averageTime="10926" name="doc-process">
      <stage averageTime="0" name="/FeedOneExample/StandardPipeManager/PerformanceLogger"/>
      <stage averageTime="5595" name="/FeedOneExample/StandardPipeManager/FetchUrl"/>
      <stage averageTime="5330" name="/FeedOneExample/StandardPipeManager/ExtractText"/>
      <stage averageTime="0" name="/FeedOneExample/StandardPipeManager/ExtractDomain"/>
      <stage averageTime="0" name="/FeedOneExample/StandardPipeManager/PrintToFile"/>
    </pipeline>
  </pipelineManager>
</executionPath>
</executionPaths>
```