

# Domino Log Analysis - Overview

## Introduction

The *Domino Log Analysis* sensor helps you to get relevant information out of the log.nsf without the need of accessing the Database. So GreenLight takes the log information and brings you to the GreenLight Frontend (or to use them for notifications).

This approach saves you a lot of time and it makes sure that you catch all relevant issues in your Domino environment!

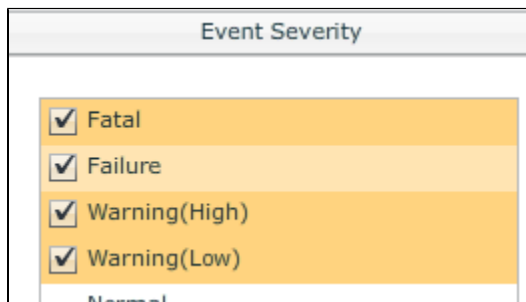
The examples below demonstrate how you can configure this Sensor for different Scenarios.

## Example

### Configuration

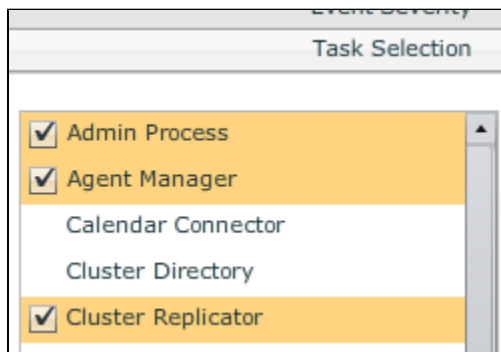
#### Monitor certain Tasks and Event Severity

- Create a *Domino Log Analysis* sensor
- Leave the *Search String* empty and expand the *Event Severity*
- Select all severity levels except *Normal* and *Unknown*



- Expand *Task Selection* and choose all the Tasks you are using on the Domino Server

Example:



- Choose your *Targets* and *Schedule* and *Save & Close* the Sensor

Go to the Live Monitor, double click on a server (where this sensor is active) , double click on the Sensor name of the Log Analysis. You should see the following screen:

Measurement Details

Statistics

Task	Warning High [1]
Router [1]	1

Click on any cell in matrix above to filter result list below

Log Entries

Details

Log Entries	Open Document

Close

In that example, the Router task reported one issue (between the measurement cycle!). Simply click on the reported issue in order to get more details. You can even click on the notes link to open the document in the log.nsf.

Measurement Details

Statistics

Task	Warning Hi
Router [1]	1

Click on any cell in matrix above to filter

Log Entries

Details

Log Entries	Open
31.03.2014 13:32:34 Router: Message 003F67D4 contains no recipients	

In case you activate also *Normal* and *Unknown* as additional Severity Types, the picture changes slightly.

Measurement Details	Statistics	
Task	Warning Low [4]	Normal [163]
SMTP Server [80]		80
Admin Process [3]		3
Router [11]		11
Agent Manager [73]	4	69

The Schedule plays an important part here. If you define a schedule of every 5 minutes, then the Log Analysis will show you only events from the "last" 5 minutes. It will not combine all previous results together!

So this was just the start. Let's continue with a more specific case.

#### Log a "Corrupt" event and notify the Admin

- Create a Domino Log Analysis sensor
- Enter *Corrupt* in the *Search String* empty and Select the first 4 Event Severities
- Expand *Task Selection* and choose all the Tasks you are using on the Domino Server
- Define the Targets and the Schedule
- Add an additional Action setting (e.g. SMTP)
- Leave the default Condition and Notification text

**Conditions**

☒ Run this action when ...

Number of messages found
 >
 0

Templates
 Custom ...

Short Text
 Log Search on \${config.nodeDefinition.name}

Text
 Total messages found: \${result.details['...']}  
 Matching Log lines:  
 (Attention: Only up to 100 of the first matching log lines will be displayed)  
 \${gl:joinWithNewLine(result.details['greenlight.logsearch.loglines'])}

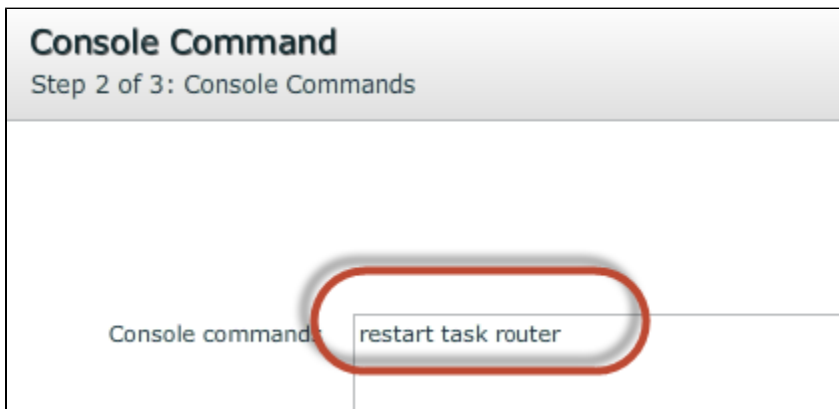
Expr. Templates

- Select and *Mail Profile* and *Save & Close* the Sensor

The Outcome is a notification whenever a Corrupt DB , etc.. is being logged to the log.nsf.

You can even trigger a Domino console command based on a log.nsf event. In bellows example a “restart task router” is issued, whenever the Router Task reports “critical events” to the log.

(Of course the event needs to be specified more precisely)



The image shows a screenshot of a web-based configuration interface for Domino. At the top, there is a header bar with the text "Console Command" in bold and "Step 2 of 3: Console Commands" below it. The main area of the interface is a large white rectangle. In the center of this area, there is a text input field. The label "Console command" is positioned to the left of the input field. The text "restart task router" is entered into the input field. A red oval is drawn around the input field, highlighting the text.

## Conclusion

This Sensor supports you in not spending too much time for reading the log.nsf ( for reading unimportant information). With this Sensor, every single event in the log.nsf can be used to trigger whatever type of action. With the console command action for instance you can start a kind of “self-healing” approach where the system tries to solves the issue automatically, based on your settings.

All in all the sensor supports you in your daily business.