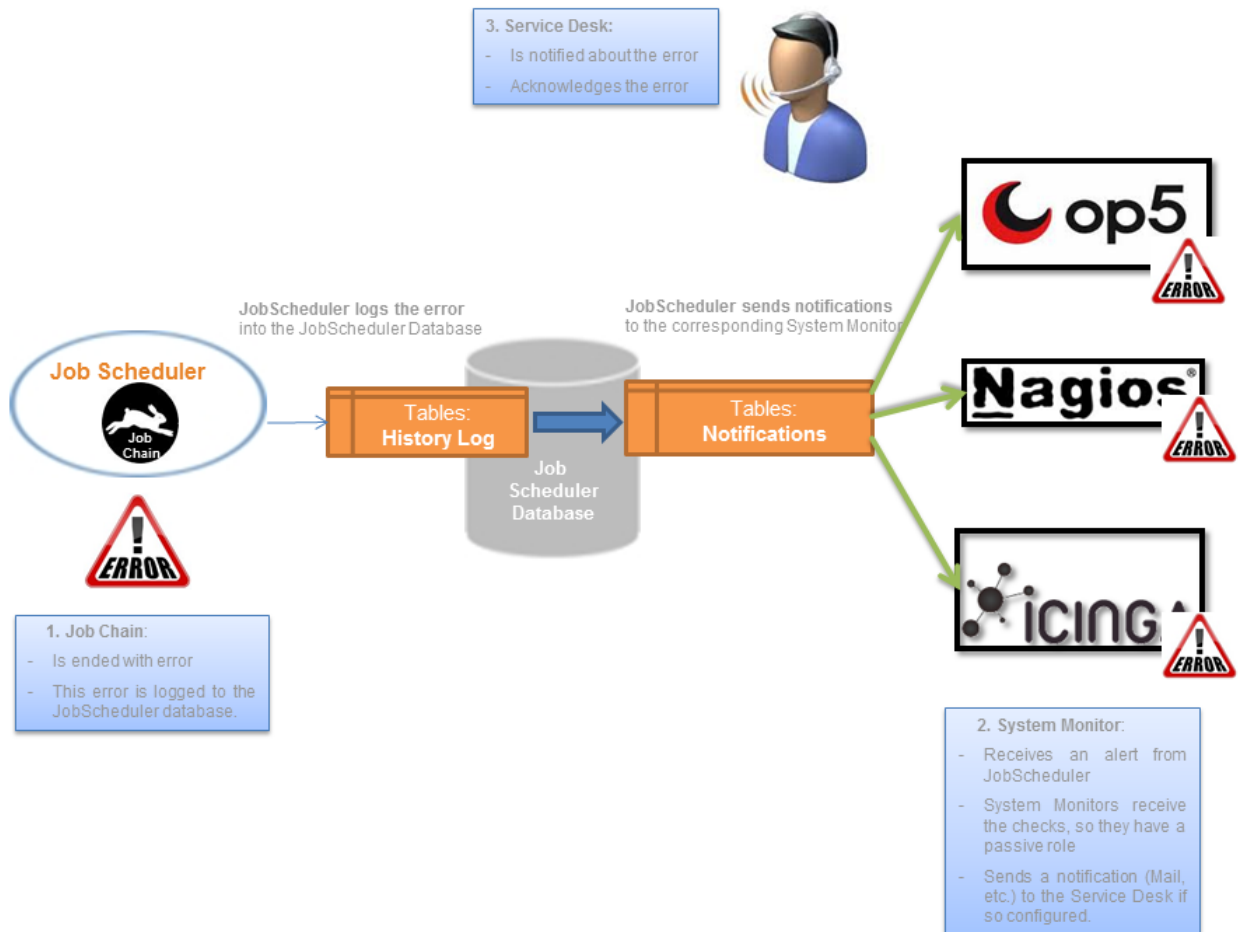


JobScheduler Monitoring Interface - Overview

- [Introduction](#)
- [Monitoring Definitions](#)
- [Benefits](#)
- [Functionality](#)
- [Monitoring example - op5® Monitor](#)
- [Change Management References](#)
- [See also](#)

Introduction

This article gives an overview of the JobScheduler Monitoring Interface. The Monitoring Interface provides an efficient means for monitoring JobScheduler objects such as [Jobs](#), [Job Chains](#) and [Orders](#) and forwarding notifications to System Monitors such as Nagios®. This solution is available with JobScheduler General Availability Release 1.8 onwards.



The most important features of this solution are:

- JobScheduler: carries out a two step process around the interface:
 - Detecting errors and other events: A Job running at regular intervals - typically every 2 minutes - analyses the History Log information recorded by the JobScheduler in the database. This job is configured not only to filter but also to analyze the log information for the Job Scheduler objects being monitored. The information noted is typically whether tasks have been successfully completed or whether errors or warnings have been logged. This job then writes this information in a separate Notifications database table.

[JITL-166](#) - Getting issue details...

STATUS

- Sending alerts: A second Job is responsible for sending the alerts to the relevant System Monitor. This job is also run at regular intervals, analyzing the Notifications database tables. It then carries out a predefined action for each item it finds in the table. Typical actions would be informing a particular monitor that a particular type of event has occurred, such as the successful completion of an order, a job ending in error or whether error recovery is being attempted.
- JobScheduler: The solution architecture allows analysis of the Log History of more than one JobScheduler using the database specified. It may also be configured to monitor more than one database.
- System Monitors: the JobScheduler is able to connect to more than one System Monitor at the same time.

Monitoring Definitions

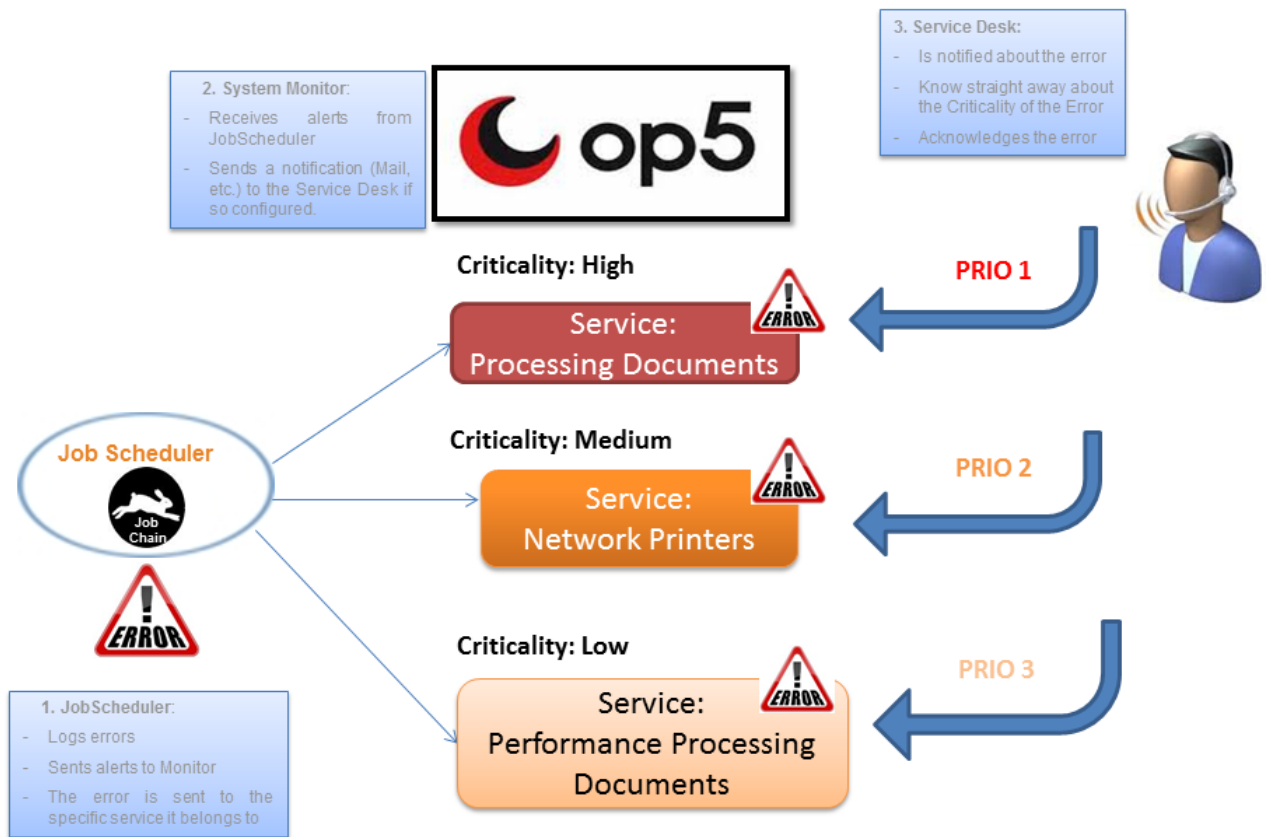
The following definitions apply for the monitoring systems:

Definition	Description
System Monitor	A System Monitor is an instrument to inform a Service Desk (e.g. 1st Level Support) about incidents in IT systems. It does not analyze incidents, but merely information about incidents, in order to be able to forward and scale this information.
Passive Checks	Passive Checks are sent remotely from an external host (from the point of view of the System Monitor) to the Monitor. Otherwise, checks that are carried out periodically by the System Monitor itself are called Active Checks.
Active Checks	Active Checks are initiated from the System Monitor server and are performed on a regular basis, e.g. every 5 minutes. They are intended for simple verification of availability of a daemon/service, they do not provide information at application level, e.g. on the execution status of jobs. Use of Active Checks is explained in How to perform active checks with a System Monitor such as Nagios/op5
Alerting	An Alert is a message about an event. An Alert does not provide all the information about an event, but it informs about the existence of the event. An Alert can be either positive or negative.
Notification	The notification of a specific Alert. Notification will not be provided for every Alert, just the ones that are so configured will be notified. Notifications are therefore a subset of the Alerts and can also be either positive or negative.
Acknowledgment	Is the confirmation of an alert and it has the meaning that the alert has been seen and/or is known and that appropriate action is being taken. An acknowledgment is always manually executed. This means that there is always someone that has realized there is a Critical service and this person acknowledges the services (usually by the Service Desk or 1st Level Support). It is never an automatized step.

Benefits

The benefits of the new solution are:

- 1. Flexible implementation:**
Changes to your existing JobScheduler configuration (Jobs, Job Chains, etc.) are not required to get this solution working. You add the Job Chains required for the monitoring but do not have to modify your current ones.
- 2. Monitor independence:**
The whole architecture lies on the JobScheduler side and the solution is therefore independent of the monitor that the Alerts are sent to. The solution works for every monitor that can receive passive checks.
- 3. Workload-independent:**
Processing of Jobs and Job Chains in JobScheduler is not affected or modified by the monitoring, neither from the point of view of performance nor that of stability.
- 4. Clearly defined information flow:**
This solution allows the information to be made available to the System Monitors to be exactly configured. Detailed log information from monitored Job Chains can be sent as a Passive Check to the relevant Monitoring Service if required.
- 5. Error Prioritization**
Errors of a critical nature are immediately recognized in the System Monitor. The JobScheduler has initially access to all the log information and can be configured to filter this information very exactly before forwarding it to the relevant System Monitor Service. This feature allows the Service Desk to be able to set priorities immediately when, for example, recovering errors: it is unlikely that a performance error would be given the same priority as an error in document processing. This feature is illustrated in the following diagram:



Functionality

Functionality	Description
Job Chain and Order Monitoring	This solution allows Job Chains to be monitored by way of the Orders that trigger these Job Chains.
History Notifications	Not only can critical alerts be monitored, but also positive ones. The history of a specific service can be monitored to see exactly if a specific work-flow has been executed and what result it gave.
Performance measurement (Timer)	Timers can be used to measure the performance of Job Chains. These can be used to send a warning alert to a System Monitor if a Job Chain takes more that a predefined time to complete.
Acknowledgment	Acknowledgments sent in response to critical alerts sent out by a System Monitor can be used to add Orders to the JobScheduler, so that the JobScheduler does not send more notifications about a service to the System Monitor.

Monitoring example - op5@ Monitor

The following example illustrates use of the JobScheduler Monitoring interface with the op5@ Monitor. In the example, three checks (in op5@ Monitor they are called services) have been defined for the JobScheduler monitoring. Different Job Chains in JobScheduler can send notifications to the same check, so that it is not necessary to create checks for each individual Job Chain, which could become extremely complex. Instead, results have been grouped in three categories:









- **JobScheduler Monitoring Errors:** Job Chains that end with an error are sent to this service. The last error notification is shown in the column "STATUS INFORMATION".
- **JobScheduler Monitoring Success:** Job Chains that end with success, that is with a positive notification, are sent to the monitoring system. To be exact, the history of a specific Job Chain is monitored to see whether a specific work-flow has been executed or not. The last success notification is shown in the column "STATUS INFORMATION".

- JobScheduler Monitoring Performance:** Here timers are used to measure the performance of a Job Chain. If a Job Chain takes too long to end, a warning alert will be sent to the System Monitor. The information about the expired timer is shown in the column "STATUS INFORMATION".

HOST	SERVICE	ACTIONS	LAST CHECK	DURATION	ATTEMPT	STATUS INFORMATION
gollum.sos	JobScheduler Monitoring Errors		2014-12-19 15:07:40	2m 4s	1/1	ERROR scheduler id=scheduler_distributed_sos, history id=1448058, job_chain=distributed/monitoring/ErrorSuccess(Error), step =100, error=SCHEDULER-280 Process terminated with exit code 127 (Dx7F)
	JobScheduler Monitoring Performance		2014-12-19 15:07:41	2m 3s	1/1	TIMER name = performance, scheduler id=scheduler_distributed_sos, history id=1448060, job_chain=distributed/monitoring/Performance (Check), check = execution time 5.00s is greater than the defined maximum time 2.00s. order started at 2014-12-19 14:05:47(UTC) and finished at 2014-12-19 14:05:52(UTC)
	JobScheduler Monitoring Success		2014-12-19 15:07:39	2m 5s	1/1	scheduler id=scheduler_distributed_sos, history id=1448059, job_chain=distributed/monitoring/ErrorSuccess(Success), order time elapsed = 0s, step =100
	System PING		2014-12-19 15:08:15	2d 18h 47m 7s	1/3	OK - 192.11.0.120: rta 1.145ms, lost 0%

Change Management References

T	Key	Linked Issues	Fix Version /s	Status	P	Summary	Updated
+	JS-1837	JITL-522 , JS-1312 , JITL-579	1.12.9	RELEASED	🔴	Send e-mail notifications via JobScheduler Monitoring Interface	Nov 18, 2019
+	JS-1600		1.9.11, 1.10.4, 1.11	RELEASED	🔵	Monitoring Interface is capable of routing Return Codes	Apr 22, 2016
+	JS-1446	JS-1445	2.0	CLARIFY	🔵	Log entries that are not dependent from JobScheduler objects should be available for JobScheduler Monitoring Interface	Dec 08, 2017
+	JS-1388		1.12	RELEASED	🔴	Add e-mail as a monitor service for the JobScheduler Monitoring Interface	Dec 21, 2017
+	JS-684	JS-1410 , JS-1589 , JS-1426 , JS-1291 , JS-1480	1.10	RELEASED	🔴	System Monitor (Nagios, op5) should notify if a JobScheduler Universal Agent is not available	Feb 10, 2016
+	JOC-731	JOC-797 , JOC-789 , JITL-579	1.13.1	RELEASED	🔴	Migrate XML Editor to JOC Cockpit	Dec 10, 2019
🔴	JITL-579	JITL-522 , JOC-731 , JS-1837	1.13.2	RELEASED	🔴	JobScheduler Monitoring Interface should process the Master messages configured with JOC Cockpit	Dec 10, 2019
🔴	JITL-567	JITL-534	1.12.10	RELEASED	🔵	FactPlugin does not set a trigger and notification order end property if an order is resumed	Aug 15, 2019
🔴	JITL-565		1.12.10	RELEASED	🔵	Monitoring Interface - Timer Minimum functionality is broken	Aug 15, 2019
🔴	JITL-564		1.12.10	RELEASED	🔵	SystemMonitor Notification generates notification for delay even if the Job chain gets fail before the maximum time of Timer	Aug 15, 2019
🔴	JITL-563		1.12.10	DISMISSED	🔵	SystemMonitor does not generate notification when the job /job chain name contains special characters like "(, @"	Jul 25, 2019
🔴	JITL-562		1.12.10	RELEASED	🔵	The entry for the column name "ORDER_END_TIME" is Null whenever the job chain fails on the node except the first and last node	Aug 15, 2019
+	JITL-534	JITL-594 , JITL-533 , JITL-567	1.12.10	RELEASED	🔵	JobScheduler Monitoring Interface - Send notifications for repeatedly failed job chain steps in JobScheduler releases starting from 1.12	Jan 23, 2020
+	JITL-533	JITL-534		DISMISSED	🔵	JobScheduler Monitoring Interface - Send notifications for repeatedly failed job chain steps in JobScheduler release 1.10	Jul 10, 2019
🔴	JITL-531	JITL-488	1.12.8	RELEASED	🔵	JobScheduler Monitoring Interface should ignore File Sink	Dec 13, 2018
+	JITL-	JS-1819 , JITL-579 , JS-	1.12.9	RELEASED	🔴	Report jobs to the JobScheduler Monitoring Interface that run	Nov 18,

522	1837				shorter/longer than expected	2019
	JITL-516	JITL-485	1.10.11, 1.12.8	RELEASED	 Notification sometimes does not notify errors when order is suspended	Feb 27, 2019
	JITL-435		1.11.6, 1.12	RELEASED	 JobScheduler Monitoring Interface should handle next steps after sending the recovery message	Jan 30, 2018
	JITL-427		1.11.6, 1.12	RELEASED	 JobScheduler Monitoring Interface. Performance improvement for SystemNotifier job.	Feb 01, 2018
	JITL-401		1.11.5, 1.12	RELEASED	 JobScheduler Monitoring Interface should support the job timers	Nov 11, 2017

Showing 20 out of 28 issues

See also

- [JobScheduler Monitoring Interface - Prerequisites and Installation](#)
- [JobScheduler Monitoring Interface - Configuration and Use Cases](#)
- [JobScheduler Monitoring Interface - XML Configuration](#)