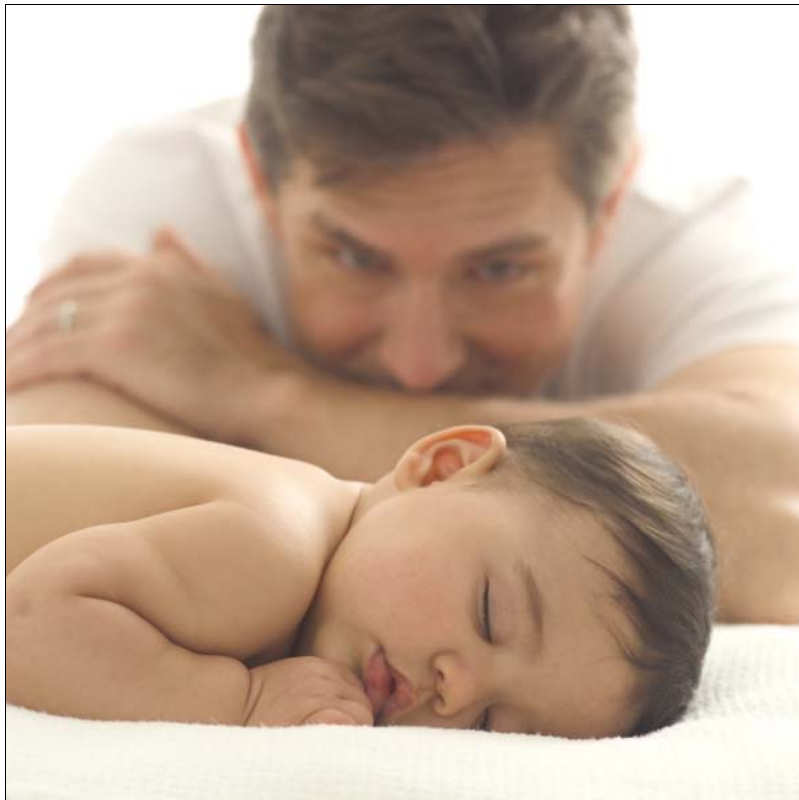


ESA System Service Monitoring

Ericsson SNMP Agent 16.0

SYSTEM ADMINISTRATION GUIDE



Copyright

© Ericsson AB 2004-2016. All Rights Reserved.

Disclaimer

No part of this document may be reproduced in any form without the written permission of the copyright owner.

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing.

Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

Ericsson is the trademark or registered trademark of Ericsson AB. All other products or service names mentioned in this document are trademarks of their respective companies.



Contents

1	About This Document	1
1.1	Purpose	1
1.2	Target Group	1
1.3	Prerequisites	1
1.4	Typographic Conventions	1
2	Introduction	3
3	System Service Monitor	5
3.1	Overview	5
3.2	Log File Scanning	5
3.3	Process Monitoring	6
3.4	System Resource Monitoring	8
3.5	Timer	9
4	Appendix A: SSM MIBs	11
	Glossary	13
	Reference List	15





1 About This Document

1.1 Purpose

The purpose of this document is to describe the system administration tasks for the Ericsson SNMP Agent (ESA).

More specifically this document is about setting up the module System Service Monitor (SSM) for monitoring the vital components and processes in a system. This document contains a very brief description of the function, it states a few useful scenarios and gives a few examples that show how easy it is to setup the monitoring. For a complete set of monitoring capabilities, see Reference [2] and Reference [3].

1.2 Target Group

The target group for this document is personnel responsible for administration of the ESA.

1.3 Prerequisites

It is assumed that the user of this document fulfils the following prerequisites.

- Has system administrator authority to the server, in which the ESA is installed.

1.4 Typographic Conventions

The typographic conventions used in this document are described in Reference [1].





2 Introduction

The ESA comes with an optional module called IBM Netcool System Service Monitor (SSM). The SSM is a monitoring feature in the system that captures system information from data sources provided by the system itself. The system does not need to communicate with the monitor processes as it is gathering information on its own.

This is very useful for systems where a monitoring solution does not exist or monitoring capabilities are not enough. By simply configuring the SSM the monitoring of the system is enhanced without affecting the system processes and operations.

The SSM is handling the system monitoring, such as capturing hardware information, scanning log files and monitoring processes.

Note: The SSM does not support the IPv6 protocol.

Caution!

Be aware of that the SSM is an optional component to the ESA and needs to be specifically ordered in order to be used.

The SSM comes with an own set of documentation, which in more detail describe the monitoring configuration. This document describes the most commonly used monitoring, while the SSM documentation describes all the monitoring capabilities available. Since the document mass is quite extensive for new ESA and SSM users, the intention with this document is to highlight the most important and useful features and lower the threshold for setting up SSM monitoring.





3 System Service Monitor

3.1 Overview

The monitoring capabilities of the SSM is huge. There are however a few monitoring function used more frequently than others.

- Log File Scanning
- Process Monitoring
- System Resource Monitoring
- Timer

This document contains a very brief description of the function, it states a few useful scenarios and gives a few examples that show how easy it is to setup the monitoring. For a complete set of monitoring capabilities, see Reference [2] and Reference [3].

3.2 Log File Scanning

3.2.1 Introduction

Description taken from the SSM documentation:

- *The logmonx subagent and the associated logMonX MIB module monitor log files for specific expressions and generate an event and notification upon finding those expressions. Using logmonx you can monitor multiple log files whose entries span multiple lines. You can also use the subagent to monitor Windows system, application and event logs.*

The Log File Scanning function could be said works as a “File to SNMP converter”. Software components that does not provide SNMP functionality built into the component normally have some kind of log file output. The ESA can simply be configured to monitor the log file and send SNMP alarms when error string matches are found. This kind of solution creates a loosely coupled relation between the component being monitored and the ESA. The component is not interfered with SNMP at all and can continue the normal behavior working towards log files.

For more information see document Reference [3].

3.2.2 Example

The log file monitor would match the following entry.



```
Fatal NI connect error 12547, connecting to:
*****
(LOCAL=NO)
VERSION INFORMATION:
  TNS for Solaris: Version 9.0.1.0.0 - Production
  Oracle Bequeath NT Protocol Adapter for Solaris:
    Version 9.0.1.0.0 - Product
  TCP/IP NT Protocol Adapter for Solaris:
    Version 9.0.1.0.0 - Production
Time: 18-JUN-2002 10:44:58
Tracing not turned on.
Tns error struct:
  nr err code: 0
  ns main err code: 12547
  TNS-12547: TNS:lost contact
  ns secondary err code: 0
  nt main err code: 0
  nt secondary err code: 0
  nt OS err code: 0
```

The following configuration lines create a trap indicating file scanning match:

```
subagent load rmonc
event reset
event type=snmp-trap
event description="Change in log /opt/oracle/sqlnet.log"
event create
lmevent=$?
```

The following configuration lines define the log file scanning:

```
subagent load logmonx
logmonx reset
logmonx logfile=/opt/oracle/sqlnet.log
logmonx filter=.*
logmonx begindelimiter="\{*8,}"
logmonx enddelimiter="nt OS err code:"
logmonx windowsize=60
logmonx updateinterval=60
logmonx event=$lmevent
logmonx eventstatus=eventReady
logmonx defaultlevel=information
logmonx create
```

3.3 Process Monitoring

3.3.1 Introduction

Description taken from the SSM documentation:



- *The process subagent provides facilities to monitor, start, stop and restart applications and processes running on a server or other network device. It can monitor multiple attributes of a running process, including status, memory usage, size and resident set size, (RSS), process time, threads, and disk and network I/O.*

The subagent identifies all processes running on the host machine and enables you to create threshold monitors for process attributes, such as CPU or memory usage, and generate an event or execute a command when a threshold is violated. By monitoring the response time, average throughput (bytes per second), as well as how much memory, CPU and network traffic each process or application uses, you can gather critical information about quality of service (QoS) and capacity planning.

With the Process Monitoring function the possibility to monitor the system processes is made very simple as the ESA can be configured to monitor the processes without interrupting or interfering with the processes. When a process is down an SNMP alarm may be sent. This kind of solution creates a soft relation between the process being monitored and the ESA as the process does not even know it is being monitored and can continue to work as it is intended to. Also, the function can not only send an alarm when a process is down. It can also try to restart the process and send a SNMP alarm clear if the restart is successful.

For more information see document Reference [3].

3.3.2

Example

The following example is monitoring a database process named "ora_pmon_XYZ" and sends an *alarm raise* when it is killed.

The following configuration lines create a trap indicating process is killed:

```
subagent load rmonc
event reset

event descr = "Process has been terminated"
event type  = snmp-trap
event create
eventprocessdown = $?
```

The following configuration lines define the process monitoring:

```
subagent load process
process reset
process description="Process XYZ is down"
process filtertype=command
process filter=.*ora_pmon_XYZ.*
process interval=10
process samplotype=exact
process attr=alive
process oper=eq
process thresh=0
```

```
process actioncmd=""
process actionevent=$eventprocessdown
process actioneventstatus=alwaysReady
process create
```

Also, the process can be monitored to send an *alarm clear* when it is restarted.

The following configuration lines create a trap indicating process is restarted:

```
subagent load rmonc
event reset
event descr = "Process is running"
event type = snmp-trap
event create
eventprocessup = $?
```

Replace the corresponding lines with the following ones:

```
process description="Process XYZ is running"
process oper=eq
process thresh=1
process actionevent=$eventprocessup
```

3.4 System Resource Monitoring

3.4.1 Introduction

Description taken from the SSM documentation:

- *The sysres subagent extends the functionality of host resource management to include additional data about CPU, memory and disks. It allows you to monitor system and application log files and gather data on system and application events.*

The sysres subagent and the associated sysRes MIB module collate and report system information. General system data is stored under the group srSystem, while other data is gathered and stored in tables. The level of system information conforms to RFC 1514/2790, as well as providing additional system information reporting.

With the System Resource Monitoring function it is easy to setup monitoring of system resources in the system. Having monitoring setup on resources, such as CPU load, Memory usage and Disk usage, using the ESA gives you the monitoring that is always requested for live systems. Also, it is possible to define thresholds to the monitoring. When a resource usage breaches a threshold a SNMP alarm may be sent.

For more information see document Reference [3].



3.4.2 Example

The following configuration lines create two traps indicating threshold is breached (violated) as well as restored (nominal):

```
subagent load rmonc
event type=snmp-trap
```

```
event descr="Metric threshold violation"
event create
eventviolation=$?
```

```
event descr="Metric threshold nominal"
event create
eventnormal=$?
```

The following configuration lines define the monitoring of the CPU load:

```
genalarm vardescr="CPU Usage Warning"
genalarm var="$srSystemCPUUsage.0"
genalarm type=absolute                mode=singleEdge
genalarm risethresh=80                fallthresh=30
genalarm riseduration=300             fallduration=0
genalarm riseevent=$eventviolation   fallevent=$eventnormal
genalarm risedescr="CPU usage above $$7% the last 5 min
    (currently $$6%)"
genalarm falldescr="CPU usage has returned to normal
    (less than $$7%)"
genalarm create
```

Similar configuration is used for monitoring of for example Disk usage and Memory usage.

3.5 Timer

3.5.1 Introduction

Description taken from the SSM documentation:

- ☐ The timer subagent MIB module provide a mechanism for generating events at a scheduled point in time.

This could be useful for, for example, setting up heartbeats from the ESA. The timer can be defined to send an event each x seconds, which will cater as heartbeat information from the system and the ESA.

For more information see document Reference [3].

3.5.2 Example

The following example makes the SSM generate an event each 5 minutes.



The following configuration lines create a trap indicating generated 5-minute event:

```
subagent load rmonc
event reset
event type=snmp-trap
event community=public
event description="Scheduled event: 5-minute intervals"
event create
schedEvent=$?
```

The following configuration lines define the timer:

```
subagent load timer
timer reset
timer minute5=0
timer event=$schedEvent
timer eventstatus=3
timer create
```



4 Appendix A: SSM MIBs

The MIBs that come with the SSM can be found in the following directory.

- `{ssm basedir}/mibs/`





Glossary

Glossary

ESA Glossary of Terms and Acronyms,
0033-CSH 109 532





Reference List

- [1] *ESA Library Overview*
DIRECTIONS FOR USE, 1/1553-CSH 109 532
- [2] *IBM Netcool/System Service Monitors Version 4.0.1, Administration Guide*
- [3] *IBM Netcool/System Service Monitors Version 4.0.1, Reference Guide*