About Ash Giddings

With more than 20 years’ experience in the IT industry and coming from an operational background, Ash Giddings is a commercially astute technical manager. He has worked for some of the largest Data Centers in Europe and in the US has advised Fortune 500 companies on major projects to save costs and improve efficiencies.

His apprenticeship was served on the IBM Mainframe, where he acquired his key skills by the following of structured processes and best practice principles.

Later in his career he applied those skills to midrange systems such as IBM i® and AIX® as well as other platforms including Windows® and Linux®.

In his role as Product Manager for Halcyon – A Division of HelpSystems, Ash enjoys finding solutions for challenging IT problems. He has a broad, generalist approach – with deep dives into technical subjects including monitoring and automation techniques for critical services, core processes and business applications as well as the art of performance management.
Foreword
by Ash Giddings

There is a proliferation of Power Systems hardware in the market place and the hardware looks identical (regardless of whether it is hosting IBM i, AIX, Linux or Windows operating systems).

This can create the perception that the operating systems running on Power, (e.g. AIX and IBM i) must be quite “similar”. The consequence of this assumption is that technical people, experienced in managing IBM i systems, are often plunged into the world of monitoring of AIX systems and quickly realize that it is an ‘older’ interface with specialist commands that do not make sense - unless you have been brought up in a UNIX environment.

In the world of business, major commercial events, like company mergers and acquisitions, also create the scenario where AIX systems may become a critical part of your infrastructure. This means that with little prior knowledge, your IT staff, who may not have AIX skills, need to be able to monitor and manage systems which are unfamiliar, but are running applications and processes on which your business relies.

In response to the scenarios above and requests from customers, I have written this guide for three reasons:

1) To demystify the art of managing AIX systems
2) To give you a reference document on what are the important things to monitor on AIX systems and why
3) To show how to make a systems administrator’s life less stressful by adopting an appropriate monitoring solution and an approach that uses pre-configured, adaptable templates for monitoring and management of your AIX environment

The purpose of this guide is to let people know that AIX is “just another operating system” and that you do not have to become an expert in AIX to successfully manage it.

This guide is aimed at technical people, systems administrators, computer operators, but also senior IT managers who need to gain an understanding of what it is important to look for (and look after) so that they can ensure the smooth running of their business operations.
# Table of Contents

Foreword .............................................................................................................. 3  
Introduction .......................................................................................................... 5  
AIX Error Report ................................................................................................... 6  
Volume Groups .................................................................................................... 6  
Logical Volumes ................................................................................................... 6  
Filesystems .......................................................................................................... 6  
Disks .................................................................................................................... 7  
CPU, Memory and Paging Space ........................................................................ 7  
Process Monitor ................................................................................................... 8  
Server Configuration ............................................................................................ 9  
Log Files ............................................................................................................. 10  
Network .............................................................................................................. 10  
Network File System .......................................................................................... 11  
Appendix 1 – AIX Monitoring Templates ............................................................ 12  
AIX System Monitoring (Basic) Template ................................................................. 12  
AIX System Monitoring (Advanced) Template ....................................................... 18  
Appendix 2 – AIX Monitoring Template Assignment .......................................... 20  
AIX System Monitoring (Standard) Template ..................................................... 20  
AIX System Monitoring (Advanced) Template .................................................... 26
Introduction

This document aims to provide a high level overview of baseline monitoring for the IBM AIX platform. The vast majority of monitoring elements detailed in this document have been included as a Central Configuration Manager template and are now shipped as part of Network Server Suite.
AIX® Monitoring Guidelines – Best Practice

AIX Error Report

AIX error codes take the following format:

Field 1 – AIX error identifier (8 digit hexadecimal)
Field 2 – Error label
Field 3 – Error type (4 character) of values: (INFO, PEND, PERF, PERM, TEMP, UNKN)
Field 4 – Error class (1 character) of values:
   H - Hardware
   S - Software
   O - Error logger
   U - Undetermined
Field 5 – Error description

Very few AIX errors would not need to be alerted to the local administrators so it is advised that all are alerted on. Your software should poll the error log and any new entries should be alerted as standard.

From AIX 5 onwards, you can limit the number of duplicates reported within a certain time frame. Hardware failure errors, Logical Volume Manager errors are generally critical and should be flagged as such.

Although not directly attributed to the Error Report its good practice to run a dumpcheck (/usr/lib/ras/dumpcheck) once a day at a peak time (this is normally added as a cron job by default on installation of AIX). If the system dump device is not big enough for a potential machine failure an error will be generated in the Error Report.

Volume Groups

Defined Volume Group statuses should always be active and there should never be any Stale Partitions.

Check that Quorum is set to off for any mirrored volume groups.

Logical Volumes

There should never be any Stale Physical Partitions associated with Logical Volumes resigning on system.

The normal working status for Logical Volumes is open/syncd.

Filesystems

Space checking should be against percentages configured by the end user for each filesystem as monitoring depends on the size and usage of the individual filesystem.
For example, a 1 TB filesystem housing Oracle datafiles would have very different requirements from a 256Mb /var filesystem on the same server.

For normal filesystems space less than 20% free should be alerted upon.

Key Filesystems /, /tmp, /usr, /home and /var should always be resident.

The availability of sufficient Inodes can be critical and you should monitor for less than 10% being free.

Journaled Filesystems should have separate logging devices.

**Disks**

Disk activity taken over a number of samples over 80% should be alerted upon.

Verify MultiPath I/O (MPIO) path failures – the normal status should read enabled.

Any disks that have a status of missing or removed would tend to indicate an issue and should be investigated (these will also be written to errpt).

A system could potentially be I/O bound if %iowait > 25% and / or %tm_act > 70% and should be investigated.

**CPU, Memory and Paging Space**

From AIX 5.3 the processor’s simultaneous multi-threading mode should be checked and alerted upon if it is capable but **not** enabled. However, in some rare situations, threading off will increase performance.

Historically the runqueue would be an element to keep a close check (alerting if the runqueue exceeded the number of CPU multiplied by 2) but with more powerful P-Series servers the value being monitored for should be increased to suit.

Although not an error as such it’s worth checking lparstat on lpar’d machines for over utilization.

Check for sustained (multiple samples of) %usr + %sys >=95%. This would tend to indicate a very busy box that could possibly be CPU bound.

The percentage of CPU time spent waiting on disk I/O (wio time) should be measured in samples and be alerted on >25% as this indicates a potential I/O bound machine.

Paging Space Utilization should be tracked and a general rule being < 30% used means that there is probably too much page space while >70% means there is probably too little page space.
The paging of the Paging Space is absolutely critical and would indicate a shortfall of RAM. Check this by monitoring for Paging > 0.

Useful additional commands used for sampling software for memory footprints:

svmon -G provides a global report. You can see the size of memory, how much is in use and the amount that is free. It provides details about how it is being used and it also provides statistics on paging space. All numbers are reported as the number of frames. A frame is 4 KB in size.

svmon -Pt 3 displays memory usage of the top 3 memory-using processes sorted in decreasing order of memory demand.

Process Monitor
Mandatory processes that should be alerted if not running:

- syncd
- errdemon
- inetd
- biod
- cron
- qdaemon
- portmap
- writesrv

Optional processes (may not be configured to run):

- rpc.lockd
- rpc.statd
- snmpd
- aixmibd
- snmpmibd
- hostmibd
- sendmail
- sshd
Server Configuration

Configuration files – Check for modifications.

- /etc/inetd.conf
- /etc/netsvc.conf
- /etc/resolv.conf
- /etc/security/login.cfg
- /etc/inittab
- /etc/aixmibd.conf
- /etc/hostmibd.conf
- /etc/snmpd.conf
- /var/spool/cron/crontabs/root
- /etc/sendmail.cf
- /etc/ssh/ssh_config (optional)
- /etc/ssh/sshd_config (optional)
- /etc/profile
- /etc/environment

Security

Check for following lines in /etc/inetd.conf – any services not being used in the environment should be commented out. Note, they are all available by default.

- ftp
- telnet
- shell
- login
- exec
- ntalk
- daytime
- time

Check users with pwck for inconsistencies in /etc/passwd & /etc/security/passwd
Log Files

Console log /var/adm/ras/conslog (alog –o –t console)
Failed logins /etc/security/failedlogin (who -a /etc/security/failedlogin)
Error Report /var/adm/ras/er rlog (last)
Cron log /var/adm/cron/log

Conslog, failedlogin and errlog are all binary based files and must be accessed via the methods above.

Network

Network problems are by their nature very difficult to spot and normally require further investigation. The starting points are listed here and are not currently shipped as a template with the Halcyon AIX Server Manager.

netstat –m:
Look for any failures on mbuf allocation. Indicates “thewall” parameter not set high enough.

netstat –p udp:
Look for “packets dropped due to no socket” != 0 and “socket overflows” >0 indicates udp send socket parameter too low.

netstat –p tcpip:
Compare the number of packets sent to the number of data packets retransmitted. If the number of packets retransmitted is over 10-15% of the total packets sent, TCP is timing out indicating that network traffic may be too high for acknowledgements (ACKs) to return before a timeout. A bottleneck on the receiving node or general network problems can also cause TCP retransmissions.

Compare the number of packets received with the number of completely duplicate packets. If TCP on a sending node times out before an ACK is received from the receiving node, it will retransmit the packet. Duplicate packets occur when the receiving node eventually receives all the retransmitted packets. If the number of duplicate packets exceeds 10-15%, the problem may again be too much network traffic or a bottleneck at the receiving node

netstat –in:
If the Oerrs column is greater than 1% of Opkts, the send queue size for that interface may need to be increased. If Ierrs is greater than 1% of Ipkts, then memory may be a problem. The transmit queue size can be changed via SMIT or the chdev command. The mtu size can be changed by the ifconfig or chdev commands or through SMIT.

netstat -v
Look if there is any value in the S/W Transmit Queue Overflow field which would indicate a need for a larger transmit queue size.
netstat -D
The -D option of netstat displays the number of packets received (lpkts), transmitted (Opkts) and dropped (ldrops, Odrops) in the communications subsystem. The important information seen here are the dropped packets particularly with the device drivers (dd). If packets are being dropped at the device driver, then you want to increase the queue size on the device driver.

Network File System

This section of the baseline document covers the Network File System (NFS) is intended as general tips and has therefore not been configured in a standard template of the Halcyon AIX Server Manager.

nfsstat -s:
If bad calls are greater than 0 the server is rejecting RPC requests. Whenever the nfsd daemon is scheduled to run but doesn't find a packet on the NFS server queue, the nullrecv field gets incremented by one. The server may be running an excessive number of nfsd daemons – this can be checked by using the command lssrc.

badlen refers to an empty or truncated RPC packet. The packet could have been damaged by a network problem.

xdrcall refers to an XDR header that may have been damaged. This is rare, but can happen more often if the network is a WAN rather than a LAN

For nfs clients:
badcalls indicates RPC failures due to timeouts (if a server does not respond within a timeout period) and interrupts (if a file system mount is interrupted with the intr option).
This differs from the badcalls as shown under the NFS statistics, which indicate authentication errors.

retrans indicates the number of retransmissions because no response was received for the server. If there is poor server response time, retrans will have a high number.
Appendix 1 – AIX Monitoring Templates

The Halcyon approach: Managing by automation and the use of rule-based templates

AIX is very text based (there is a graphical user interface but it is not a sophisticated one) so you can look at the AIX console and run powerful commands that give you really good information - but nevertheless it’s only displayed on a console. This means that unless you are watching it 24 x 7 you might miss something important.

A better way of doing things is to have a solution in place which can run the AIX commands, automatically ‘read’ the responses and one which “taps you on the shoulder” if there is an issue or something is about to go wrong. To cover all bases it makes sense to use the same solution to handle your escalation procedures and also send information to one console for centralized management of all alerts across your enterprise.

Look for solutions which will integrate with your helpdesk system and can also provide, via mobile ‘apps’, access to the Enterprise Console using your smartphones and tablets.

Halcyon have made the investment to develop solutions which automate processes, notify IT staff of any issues (via email, console display or mobile devices), comply with your escalation procedures, and integrate with your existing IT systems and your helpdesk software.

Working with IBM business partners, ISVs and customers Halcyon have also created rule-based templates to monitor AIX systems and a comprehensive list is presented below.

AIX System Monitoring (Basic) Template

The AIX System Monitoring (Basic) template contains filters covering all of the AIX® Monitors with the exception of the System Monitor. The following filters are defined:

AIX Error Report Monitor

This monitor checks against the AIX Error Report, which contains a list of logged errors. This monitor contains the following two filters:

Hardware Errors - Errpt(Class=H)

This filter checks for any hardware errors that are reported within the AIX Error Report. These are identified as being of class=H. An alert is generated if any class=H errors are found within the AIX Error Report.

Software Errors - Errpt(Class=S)

This filter checks for any software errors that are reported within the AIX Error Report. These are identified as being of class=S. An alert is generated if any class=S errors are found within the AIX® Error Report.
**AIX Subsystem Report Monitor**
This monitor checks for critical subsystems being present and active. It contains the following six filters:

**Critical Subsystem (inetd) Does Not Exist - Subsystem Does Not Exist(inetd)**
This filter checks for the existence of the AIX subsystem ‘inetd’. An alert is generated if this critical subsystem is not found.

**Critical Subsystem (inetd) is Inoperative - Subsystem is Inoperative(inetd)**
This filter checks that the AIX subsystem ‘inetd’ is active. An alert is generated if this critical subsystem is in an inoperative state.

**Critical Subsystem (qdaemon) Does Not Exist - Subsystem Does Not Exist(qdaemon)**
This filter checks for the existence of the AIX subsystem ‘qdaemon’. An alert is generated if this critical subsystem is not found.

**Critical Subsystem (qdaemon) is Inoperative - Subsystem is Inoperative(qdaemon)**
This filter checks that the AIX subsystem ‘qdaemon’ is active. An alert is generated if this critical subsystem is in an inoperative state.

**Critical Subsystem (syslogd) Does Not Exist - Subsystem Does Not Exist(syslogd)**
This filter checks for the existence of the AIX subsystem ‘syslogd’. An alert is generated if this critical subsystem is not found.

**Critical Subsystem (syslogd) is Inoperative - Subsystem is Inoperative(syslogd)**
This filter checks that the AIX subsystem ‘syslogd’ is active. An alert is generated if this critical subsystem is in an inoperative state.

**Logical Volume Monitor**
The AIX Logical Volume monitors the status of Logical Groups, Logical Volumes and Physical Volumes of the AIX system and contains the following two filters:

**Alert when Quorum is Set To On When Disk Mirroring is Active(rootvg) - Measure(Quorum) Trigger(=0)**
A quorum is a state in which 51 percent or more of the physical volumes in a volume group are accessible. A quorum is a vote of the number of Volume Group Descriptor Areas and Volume Group Status Areas (VGDA/VGSA) that are active. A quorum ensures data integrity in the event of a disk failure. When a quorum is lost, the volume group varies itself off so that the disks are no longer accessible by the Logical Volume Manager (LVM). This filter checks to ensure that the Quorum is available when disk mirroring is active and raises an alert if set to On.

**Volume Group (rootvg) Does Not Exist - Volume Group rootvg Does Not Exist**
Root Volume Group (rootvg) is a volume group containing the Base Operating System (BOS). This filter checks to ensure that the Root Volume Group exists and raises an alert if it is not found to be present.
Script Monitor
The AIX Script Monitor runs custom AIX scripts and commands and checks the output against Regular Expressions. This contains the following two filters:

Check for Failed Logins - Script(/var/lib/halcyon/logfails.sh denied)
This filter checks for any failed login attempts. Any failed login attempts found are entered in the failedlogins.log. The filter, ‘Monitor for Failed Logins’ within the Log File Monitor, then looks for any entries within this file.

Check for Missing or Removed Disks - Script(lspv missing | removed)
This filter runs the lspv command and checks to see if any disks are reported as missing or removed. An alert is raised upon discovery of either of these two conditions.

AIX File & Folder Monitor
The AIX File & Folder Monitor checks the current status of specific files within the AIX system and contains the following seven filters:

File (/etc/aixmibd.conf) Has Changed - File(/etc/aixmibd.conf) Trigger(Has Changed)
This filter checks the file; /etc/aixmibd.conf. This file is used to configure the thresholds for many AIX monitors. An alert is generated if a change is detected to the modified date of this file.

File (/etc/inetd.conf) Has Changed - File(/etc/inetd.conf) Trigger(Has Changed)
This filter checks the file; /etc/inetd.conf. This file, also known as the super server, loads a network program based upon a request from the network. An alert is generated if a change is detected to the modified date of this file.

File (/etc/inittab) Has Changed - File(/etc/inittab) Trigger(Has Changed)
This filter checks the file; /etc/inittab. This file is a script that controls most of the boot sequence. It dictates what programs and scripts to launch and at what runlevels. An alert is generated if a change is detected to the modified date of this file.

File (/etc/profile) Has Changed - File(/etc/profile) Trigger(Has Changed)
This filter checks the file; /etc/profile. This file contains system wide environment details and startup programs. An alert is generated if a change is detected to the modified date of this file.

File (/etc/security/login.cfg) Has Changed - File(/etc/security/login.cfg) Trigger(Has Changed)
This filter checks the file; /etc/security/login.cfg. The /etc/security/login.cfg file is an ASCII file that contains stanzas of configuration information for login and user authentication. An alert is generated if a change is detected to the modified date of the file.

File (/etc/sendmail.cf) Has Changed - File(/etc/sendmail.cf) Trigger(Has Changed)
This filter checks the file; /etc/sendmail.cf. This is a lengthy and detailed configuration file and direct editing of this file should be avoided. An alert is generated if a change is detected to the modified date of the file.

File (/var/spool/cron/crontabs/root) Has Changed - File(/var/spool/cron/crontabs/root) Trigger(Has Changed)
The /var/spool/cron/crontabs/root file contains commands needed for basic system control. This filter checks this file and raises an alert if any changes have been made to the file since the last time the filter ran.
Log File Monitor
The AIX Log File Monitor checks AIX Log Files for failed user login attempts and new entries being posted to the Cron Log file. This monitor contains the following two filters:

Monitor for Failed Logins - LogFile(/var/lib/halcyon/failedlogins.log) Expression(.*)
This filter checks the failedlogins.log file for any entries. An alert is generated if any failed logins are reported within the /var/lib/halcyon/failedlogins.log file.

Monitor for New Entries in Cron Log - LogFile(/var/adm/cron/log) Expression(.*)
This filter checks the /var/adm/cron/log for any new entries. The cron daemon, that controls the automatic running of commands creates a log of its activities in the /var/adm/cron/log file. An alert is generated if any new entries are recorded in this log file.

CPU, Filesystem and Memory Monitor
The AIX CPU, Filesystem and Memory Monitor template contains the following sixteen filters, that measure filesystem, memory and CPU performance and alerts if thresholds are breached.

Filesystem (/) Disk Space Used >=80% - Group(Filesystem) Volume(/)
Type(UsedPercent) Trigger(>=80%)
This filter checks that the root filesystem ‘/’ on volume ‘/’ has more than 20% free space available at all times. An alert is generated if the available disk space on filesystem ‘/’ equals or exceeds 80 percent.

Filesystem (/) Does Not Exist - Group(Filesystem) Volume(/) Trigger(Does Not Exist)
This filter checks that the root filesystem ‘/’ is in existence on volume ‘/’. An alert is generated if the root filesystem ‘/’ is not found on volume ‘/’.

Filesystem (/) Inode Used >=90% - Group(Filesystem) Volume(/) Type(I-Nodes Used %) Trigger(>=90%)
This filter checks the percentage Inode used on root filesystem ‘/’. An inode is a data structure in UNIX operating systems that contains important information pertaining to files within a file system. When a file system is created in UNIX, a set amount of inodes are also created. Usually, about 1 percent of the total file system disk space is allocated to the inode table. An alert is generated if the percentage Inode used on root filesystem ‘/’ equals or exceeds 90 percent.

Filesystem (/home) Disk Space Used >=80% - Group(Filesystem) Volume(/home)
Type(UsedPercent) Trigger(>=80%)
This filter checks that the filesystem ‘/home’ on volume ‘/home’ has more than 20% free space available at all times. An alert is generated if the available disk space on filesystem ‘/home’ equals or exceeds 80 percent.

Filesystem (/home) Does Not Exist - Group(Filesystem) Volume(/home) Trigger(Does Not Exist)
This filter checks that the filesystem ‘/home’ is in existence on volume ‘/home’. An alert is generated if the root filesystem ‘/home’ is not found on volume ‘/home’.

Filesystem (/home) Inode Used >=90% - Group(Filesystem) Volume(/home) Type(I-Nodes Used %) Trigger(>=90%)
This filter checks the percentage Inode used on filesystem ‘/home’. An alert is generated if the percentage Inode used on root filesystem ‘/home’ equals or exceeds 90 percent.
Filesystem (/tmp) Disk Space Used >=80% - Group(Filesystem) Volume(/tmp) 
Type(UsedPercent) Trigger(>=80%)
This filter checks that the filesystem ‘/tmp’ on volume ‘/tmp’ has more than 20% free space available at all times. An alert is generated if the available disk space on filesystem ‘/tmp’ equals or exceeds 80 percent.

Filesystem (/tmp) Does Not Exist - Group(Filesystem) Volume(/tmp) Trigger(Does Not Exist)
This filter checks that the filesystem ‘/tmp’ is in existence on volume ‘/tmp’. An alert is generated if the root filesystem ‘/tmp’ is not found on volume ‘/tmp’.

Filesystem (/tmp) Inode Used >=90% - Group(Filesystem) Volume(/tmp) Type(I-Nodes Used %) Trigger(>=90%)
This filter checks the percentage Inode used on filesystem ‘/tmp’. An alert is generated if the percentage Inode used on root filesystem ‘/tmp’ equals or exceeds 90 percent.

Filesystem (/usr) Disk Space Used >=80% - Group(Filesystem) Volume(/usr) 
Type(UsedPercent) Trigger(>=80%)
This filter checks that the filesystem ‘/usr’ on volume ‘/usr’ has more than 20% free space available at all times. An alert is generated if the available disk space on filesystem ‘/usr’ equals or exceeds 80 percent.

Filesystem (/usr) Does Not Exist - Group(Filesystem) Volume(/usr) Trigger(Does Not Exist)
This filter checks that the filesystem ‘/usr’ is in existence on volume ‘/usr’. An alert is generated if the root filesystem ‘/usr’ is not found on volume ‘/usr’.

Filesystem (/usr) Inode Used >=90% - Group(Filesystem) Volume(/usr) Type(I-Nodes Used %) Trigger(>=90%)
This filter checks the percentage Inode used on filesystem ‘/usr’. An alert is generated if the percentage Inode used on root filesystem ‘/usr’ equals or exceeds 90 percent.

Filesystem (/var) Disk Space Used >=80% - Group(Filesystem) Volume(/var) 
Type(UsedPercent) Trigger(>=80%)
This filter checks that the filesystem ‘/var’ on volume ‘/var’ has more than 20% free space available at all times. An alert is generated if the available disk space on filesystem ‘/var’ equals or exceeds 80 percent.

Filesystem (/var) Does Not Exist - Group(Filesystem) Volume(/var) Trigger(Does Not Exist)
This filter checks that the filesystem ‘/var’ is in existence on volume ‘/var’. An alert is generated if the root filesystem ‘/var’ is not found on volume ‘/var’.

Filesystem (/var) Inode Used >=90% - Group(Filesystem) Volume(/var) Type(I-Nodes Used %) Trigger(>=90%)
This filter checks the percentage Inode used on filesystem ‘/var’. An alert is generated if the percentage Inode used on root filesystem ‘/var’ equals or exceeds 90 percent.

Sustained CPU >95% - Group(CPU) CPU(0) Type(Load) Trigger(>95%)
This filter checks the sustained usage of the CPU. An alert is generated if the sustained CPU load exceeds 95% at any one time.
AIX Process Monitor

The AIX Process Monitor is used to check that critical processes exist on the system. This monitor contains the following eight filters:

**Critical Process (biod) Does Not Exist - Type(Process By Name) Process(biod)**

Trigger(DoesNotExist)

The biod daemon is required on systems that are either mounting (as a client) or exporting (as a server) filesystems via NFS. This filter checks that the critical process, biod, exists on the system and generates an alert if it is not present.

**Critical Process (cron) Does Not Exist - Type(Process By Name) Process(cron)**

Trigger(DoesNotExist)

The cron daemon runs shell commands at specified dates and times. This filter checks that the critical process, cron, exists on the system and generates an alert if it is not present.

**Critical Process (errdemon) Does Not Exist - Type(Process By Name) Process(errdemon)**

Trigger(DoesNotExist)

This critical process starts the error logging daemon (errdemon) and writes entries to the error log. This filter checks that this critical process exists on the system and generates an alert if it is not present.

**Critical Process (inetd) Does Not Exist - Type(Process By Name) Process(inetd)**

Trigger(DoesNotExist)

The inetd critical process provides Internet service management for a network. This filter checks that the inetd process exists on the system and generates an alert if it is not present.

**Critical Process (portmap) Does Not Exist - Type(Process By Name) Process(portmap)**

Trigger(DoesNotExist)

The portmap daemon helps clients map program number and version number pairs to the port number of a server. This filter checks that the portmap process exists on the system and generates an alert if it is not present.

**Critical Process (qdaemon) Does Not Exist - Type(Process By Name) Process(qdaemon)**

Trigger(DoesNotExist)

The qdaemon tracks both job requests and the resources necessary to complete the jobs, where the resources may be a real printer, some other real device, or even a file. This filter checks that the qdaemon process exists on the system and generates an alert if it is not present.

**Critical Process (syncd) Does Not Exist - Type(Process By Name) Process(syncd)**

Trigger(DoesNotExist)

A file synchronization program, the syncd daemon flushes pages at regularly scheduled intervals. This filter checks that the syncd process exists on the system and generates an alert if it is not present.

**Critical Process (writesrv) Does Not Exist - Type(Process By Name) Process(writesrv)**

Trigger(DoesNotExist)

The writesrv daemon enables the ability of users on remote machines to use the ‘write’ command to send messages to users on the local system. This filter checks that the writesrv process exists on the system and generates an alert if it is not present.
**AIX Ping Monitor**

This monitor contains a single filter:

**Check Server Can Ping Router - Host(1.2.3.4) Timeout(2000) Attempts(4) Success(50%) TTL(128)**

This filter checks that the server can ping a defined user-defined router. The host address of this template needs to be amended to the actual IP address of the router that you wish to monitor. An alert is generated if the success rate is less than 50% over the four attempts that the ping makes to communicate with the router.

**AIX System Monitoring (Advanced) Template**

The AIX System Monitoring (Advanced) template contains all of the filters available in the basic template and adds additional filter coverage.

**AIX Error Report Monitor**

This contains the basic filters plus these additional two filters:

**Error Logger Errors - Errpt(Class=O)**

This filter checks for any hardware errors that are reported within the AIX Error Report. These are identified as being of class=O. An alert is generated if any class=H errors are found within the AIX® Error Report.

**Undetermined Errors - Errpt(Class=U)**

This filter checks for any hardware errors that are reported within the AIX Error Report. These are identified as being of class=U. An alert is generated if any class=U errors are found within the AIX® Error Report.

**AIX Subsystem Monitor**

This contains the basic filters plus these additional six filters that cover non-critical subsystems:

**Optional Subsystem (dhcppd) Does Not Exist - Subsystem Does Not Exist(dhcppd)**

This filter checks for the existence of the AIX subsystem ‘dhcppd’. An alert is generated if this optional subsystem is not found.

**Optional Subsystem (dhcppd) is Inoperative - Subsystem is Inoperative(dhcppd)**

This filter checks that the AIX subsystem ‘syslogd’ is active. An alert is generated if this optional subsystem is in an inoperative state.

**Optional Subsystem (lpd) Does Not Exist - Subsystem Does Not Exist(lpd)**

This filter checks for the existence of the AIX subsystem ‘lpd’. An alert is generated if this optional subsystem is not found.

**Optional Subsystem (lpd) is Inoperative - Subsystem is Inoperative(lpd)**

This filter checks that the AIX subsystem ‘lpd’ is active. An alert is generated if this optional subsystem is in an inoperative state.

**Optional Subsystem (sendmail) Does Not Exist - Subsystem Does Not Exist(sendmail)**

This filter checks for the existence of the AIX subsystem ‘sendmail’. An alert is generated if this optional subsystem is not found.
Optional Subsystem (sendmail) is Inoperative- Subsystem is Inoperative(sendmail)
This filter checks that the AIX subsystem ‘sendmail’ is active. An alert is generated if this optional subsystem is in an inoperative state.

AIX Logical Volume Monitor
This contains the basic filters plus these additional six filters:

Stale Physical Partitions Exist (/Mounted on rootvg hd4) - Measure(Stale Physical Partitions) Trigger(=0)
A STALE physical partition is a physical partition that contains data you cannot use. This filter checks to see if any stale physical partitions exist on the Root Volume Group hd4.

Stale Physical Partitions Exist (/home Mounted on rootvg hd1) - Measure(Stale Physical Partitions) Trigger(=0)
This filter checks to see if any stale physical partitions exist on the Root Volume Group hd1.

Stale Physical Partitions Exist (/tmp Mounted on rootvg hd3) - Measure(Stale Physical Partitions) Trigger(=0)
This filter checks to see if any stale physical partitions exist on the Root Volume Group hd3.

Stale Physical Partitions Exist (/usr Mounted on rootvg hd2) - Measure(Stale Physical Partitions) Trigger(=0)
This filter checks to see if any stale physical partitions exist on the Root Volume Group hd2.

Stale Physical Partitions Exist (/usr Mounted on rootvg hd9) - Measure(Stale Physical Partitions) Trigger(=0)
This filter checks to see if any stale physical partitions exist on the Root Volume Group hd9.

Note: All actions for each of the above filters within these templates are set to a default of sending an alert to the Enterprise Console. You must manually change this setting if you require an alternative action to be taken upon the generation of an alert.
Appendix 2 – AIX Monitoring Template Assignment

The following forms are provided for your use to assist you in your application and configuration of the Halcyon templates within your organization. You can print these forms for manual completion which you can then use when applying the template rules within your own organization.

Each example Rule Group is listed, together with the individual rules contained within. You can determine whether or not you wish to activate the rule, which values you wish to apply and any comments that you wish to make (such as specific inclusions or omissions for example).

### AIX System Monitoring (Standard) Template

<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX Error Report Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Errors – Errpt(Class=H)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Errors – Errpt(Class=S)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIX Subsystem Report Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Subsystem (inetd) Does Not Exist – Subsystem Does Not Exist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Subsystem (inetd) is Inoperative – Subsystem is inoperative(inetd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Subsystem (qdaemon) Does Not Exist – Subsystem Does Not Exist (qdaemon)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Subsystem (qdaemon) is Inoperative – Subsystem is Inoperative (qdaemon)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>Critical subsystem (syslogd) Does Not Exist – Subsystem Does Not Exist (syslogd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Subsystem (syslog) is Inoperative – Subsystem is Inoperative (syslogd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical Volume Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alert when Quorum is Set To On When Disk Mirroring is Active (rootvg) – Measure(Quorum) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volume Group (rootvg) Does Not Exist – Volume Group rootvg Does Not Exist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Script Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check for Failed Logins – Script(/var/lib/halcyon/logfails.sh denied)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check for Missing or Removed Disks – Script(lspv missing</td>
<td>removed)</td>
<td></td>
</tr>
<tr>
<td>AIX File &amp; Folder Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/aixmibd.conf) Has Changed – File(/etc/aixmibd.conf) Trigger(Has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>File (/etc/inetd.conf) Has Changed – File (/etc/inetd.conf) Trigger (Has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/inittab) Has Changed – File (/etc/inittab) Trigger (Has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/profile) Has Changed – File (/etc/profile) Trigger (has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/security/login.cfg) Has Changed – File (/etc/security/login.cfg) Trigger (Has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/sendmail.cf) Has Changed – File (/etc/sendmail.cf) Trigger (Has Changed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/var/spool/cron/crontabs/root) Has Changed – File (/var/spool/cron/crontabs/root) Trigger (Has Changed)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Log File Monitor**

- **Monitor for Failed Logins** – LogFile (/var/lib/halcyon/failedlogins.log) Expression (.*)

- **Monitor for New Entries in Cron Log** – LogFile (var/adm/cron/log) Expression (.*
<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU, FileSystem and Memory Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/) Disk Space Used &gt;80% - Group(Filesystem) Volume(/) Type(usedPercent) Trigger(&gt;80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/) Does Not Exist – Group(Filesystem) Volume(/) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/) Inode Used &gt;=90% - Group(Filesystem) Volume(/) Type(I-Nodes Used %) Trigger(&gt;=90%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/home) Disk Space Used &gt;=80% - Group(Filesystem) Volume(/home) Type(UsedPercent) Trigger(&gt;=80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/home) Does Not Exist – Group(Filesystem) Volume(/home) Trigger(does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/home) Inode Used &gt;=90% - Group(Filesystem) Volume(/home) Type(I-Nodes Used %) Trigger(&gt;=90%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/tmp) Disk Space Used &gt;=80% - Group(Filesystem) Volume(/tmp) Type(UsedPercent) Trigger(&gt;=80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>Filesystem (/tmp) Does Not Exist – Group(Filesystem) Volume(/tmp) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/tmp) Inode Used &gt;90% - Group(Filesystem) Volume(/tmp) Type(I-Nodes Used %) Trigger(&gt;90%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/usr) Disk Space Used &gt;=80% - Group(Filesystem) Volume(/usr) Type(UsedPercent) Trigger(&gt;=80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/usr) Does Not Exist – Group(Filesystem) Volume(/usr) Type(UsedPercent) Trigger(&gt;=80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/usr) Inode Used &gt;=90% - Group(Filesystem) Volume(/usr) Type(I-Nodes Used %) Trigger(&gt;=90%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/var) Disk Space Used &gt;=80% - Group(Filesystem) Volume(/var) Type(UsedPercent) Trigger(&gt;=80%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filesystem (/var) Does Not Exist – Group(Filesystem) Volume(/var) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>Filesystem (/var) Inode Used &gt;=90% - Group(Filesystem) Volume(/var) Type(I-Nodes Used%) Trigger(&gt;=90%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustained CPU &gt;95% - Group(CPU) CPU(0) Type(Load) Trigger(&gt;95%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AIX Process Monitor**

<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Process (biod) Does Not Exist – Type(process By Name) Process(biod) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (cron) Does Not Exist – Type(Process By Name) Process(cron) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (errdemon) Does Not Exist – Type(Process By Name) Process(cron) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (inetd) Does Not Exist – Type(Process By Name) Process(inetd) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (portmap) Does Not Exist – Type(Process By Name) Process(portmap) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## AIX® Monitoring Guidelines – Best Practice

### Description

<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Process (qdaemon) Does Not Exist – Type(Process By Name) Process(qdaemon) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (syncd) Does Not Exist – Type(Process By Name) Process(syncd) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Process (writesrv) Does Not Exist – Type(Process By Name) Process(writesrv) Trigger(DoesNotExist)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### AIX Ping Monitor

<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Server Can Ping Router – Host(1.2.3.4) Timeout(2000) Attempts(4) Success(50%) TTL(128)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### AIX System Monitoring (Advanced) Template

<table>
<thead>
<tr>
<th>Description</th>
<th>Activate?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX Error Report Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error Logger Errors – Errpt(Class=O)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undetermined Errors – Errpt(Class=U)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>AIX Subsystem Report Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Subsystem (dhcpcd) Does Not Exist – Subsystem Does Not Exist(dhcpcd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optional Subsystem (dhcpcd) is Inoperative – Subsystem is Inoperative(dhcpcd)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Subsystem (lpd) Does Not Exist – Subsystem Does Not Exist</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optional Subsystem (lpd) is Inoperative – Subsystem is Inoperative</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Subsystem (sendmail) Does Not Exist – Subsystem Does Not Exist(sendmail)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optional Subsystem (sendmail) is Inoperative – Subsystem is Inoperative(sendmail)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AIX Logical Volume Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stale Physical Partitions Exist (/Mounted on rootvg hd4) – Measure(Stale Physical Partitions) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stale Physical Partitions Exist (/home Mounted on rootvg hd1) – Measure(Stale Physical Partitions) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>Stale Physical Partitions Exist (/tmp Mounted on rootvg hd3) – Measure(Stale Physical Partitions) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stale Physical Partitions Exist (/usr Mounted on rootvg hd2) – Measure(Stale Physical Partitions) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stale Physical Partitions Exist (/usr Mounted on rootvg hd9) – Measure(Stale Physical Partitions) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stale Physical Partitions on rootvg – Measure(Stale Physical Volumes) Trigger(=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIX Script Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Console Log – Script(/var/lib/halcyon/conslog.sh.+ )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk I/O Busy &gt;75% Possible I/O Bound System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk I/O Wait &gt;25% Possible I/O Bound System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustained Disk Utilization &gt;=80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify Operations Status of Path to MPIO Device</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>AIX File &amp; Folder Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/environment) Has Changed –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File(/etc) Trigger(Exists)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File (/etc/hostmibd.conf) Has Changed –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File(/rtc) Trigger(Exists)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AIX CPU, Filesystem and Memory Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page File Used &lt;30% (Suggests Too Much Paging Space) –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group(Memory) Type(page File Used %) Trigger(&lt;30%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page File Used &gt;70% (Suggests Not Enough Paging Space) –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group(Memory) Type(page File Used %) Trigger(=70%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paging Space &gt;95% -</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group(Memory) Type(Page File Used %) Trigger(=95%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Process Monitor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (aixmidb) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(aixmbd) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (hostmibd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(hostmibd) Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Activate?</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>Optional Process (rpc.lockd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(rpc.lockd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (rpc.statd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(rpc.statd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (sendmail) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(sendmail)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (snmpd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(snmpd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (snmpmibd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(snmpmibd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Process (sshd) Does Not Exist –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type(Process By Name) Process(sshd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger(Does Not Exist)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>