

CoRS Health Check Linux

Late 2017

Status: Released for Approval

Version: 1.01

Target Release: 2017

**Prepared by:** Louis Puskas

**Date Created**: 10/24/2017

**Date Last Modified**: 10/024/2017

**Approvals**

Documented approvals are **required** from all approvers designated below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Project Role** | | **Approval Documentation** |
| **Mark Horn** | | **CoRS – Technology Manager** |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

**Contributors**

The listed individuals provided content to this document.

|  |  |
| --- | --- |
| **Name** | **Project Role** |
| Louis Puskas | **DAP Metadata Architect** |
|  |  |
|  |  |

**Version History**

Changes made during the creation and finalization of this document

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Revision Date** | **Revised By** | **Change Descriptions** |
| V1.00 | 10/24/2017 | Louis Puskas | First draft created. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Glossary of Terms**

Terms and acronyms within this document that may be unfamiliar to readers.

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| **ETL Nodes** | All of our Ab Initio development enviornmenst are made up of a network of computers which clustered to together virtually. In this document a regulary refer to one of the computers in that virtual cluster as an ETL Node. |
| **Heat Map Report** | Heat map reporting is used to determine when a piece of your infrastructure is being used heavily vs lightly.  The heat map report created by this solution will show you the history of alerts on the server it’s run for, regardless of the type of alert which was triggered. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

[1. Document Purpose 5](#_Toc496621663)

[1.1 Business Case and Objectives 5](#_Toc496621664)

[2. Project Scope & Dependencies 5](#_Toc496621665)

[2.1 In Scope 5](#_Toc496621666)

[2.2 Out of Scope 5](#_Toc496621667)

[2.3 Project Dependencies 5](#_Toc496621668)

[2.4 Project Assumptions 5](#_Toc496621669)

[2.5 Key Constraints 5](#_Toc496621670)

[3. Determining which servers to monitor 6](#_Toc496621671)

[4. Where are the script components located? 6](#_Toc496621672)

[4.1 Location of the main wrapper script and it’s per environment control files. 6](#_Toc496621673)

[4.2 Location of the individual check scripts and their per environment control files 7](#_Toc496621674)

[5. How does the script run ? 8](#_Toc496621675)

[5.1 How to start the script 8](#_Toc496621676)

[5.2 How to check if the script is running properly 8](#_Toc496621677)

[6. Where is the output located? 8](#_Toc496621678)

[7. What items are checked by the Health Alerts ? 9](#_Toc496621679)

[7.1.1 check\_daemon\_new 9](#_Toc496621680)

[7.1.2 check\_df\_new 9](#_Toc496621681)

[7.1.3 check\_dir\_perm\_new 10](#_Toc496621682)

[7.1.4 check\_EME\_status\_new 10](#_Toc496621683)

[7.1.5 check\_ipcs\_new 10](#_Toc496621684)

[7.1.6 check\_mem\_new 10](#_Toc496621685)

[7.1.7 check\_netif\_speed\_volume\_new 11](#_Toc496621686)

[7.1.8 check\_new\_rpm\_new 11](#_Toc496621687)

[7.1.9 check\_per\_user\_process\_count\_new 11](#_Toc496621688)

[7.2.1 check\_top\_new 12](#_Toc496621689)

[7.2.2 check\_vmstat\_new 12](#_Toc496621690)

# Document Purpose

The CoRS Linux Health Check alerts were installed in the 2nd quarter of 2017 by Louis Puskas to assist the CoRS ETL team in finding performance issues with their Ab Initio code.

This document will describe how the health alerts have been installed, how they work, and delve into some of the topics related to the issues reported by the health alerts.

## 

## 1.1 Business Case and Objectives

CoRS management was seeking a solution which would proactively warn them about issues in their Ab Initio shared technology environment while also gathering statistics on executions of their deployed code in controlled environments like SIT, UAT and PROD.

# Project Scope & Dependencies

## 2.1 In Scope

| **ID** | **Description** |
| --- | --- |
| 2.1.1 | Installation of health\_check\_linux suite on all ETL nodes used by the CoRS Ab Initio team. |
| 2.1.2 |  |

## 2.2 Out of Scope

| **ID** | **Description** |
| --- | --- |
| 2.2.1 | Automatic restart of the health\_check\_linux suite by the EDA system administrators after server patching or reboot. |
| 2.2.2 |  |
| 2.2.3 |  |

## 

## 2.3 Project Dependencies

| ID | Project Dependencies |
| --- | --- |
| 2.3.1 |  |
| 2.3.2 |  |

## 2.4 Project Assumptions

| ID | Assumption |
| --- | --- |
| 2.4.1 | The alerts will be sent to individuals with a basic understating of UNIX operating system concepts. This document will not delve too much into training on what constitutes a legit issue in Linux. The good news is the scripts are intelligent to identify the issues for you. What you do about it is about to you. |
| 2.4.2 |  |

## 2.5 Key Constraints

|  |  |
| --- | --- |
| **#** | **Constraints** |
| 2.5.1 | *N/A* |
|  |  |

# Determining which servers to monitor

The quickest way to determine which Linux machines are used by your Ab Initio ETL team is to query the job scheduler used by that team.

The job scheduler will give you insight into every ETL nodes it may send a new job requests to.

The job scheduler will bring back virtual hosts names. A little work will be done after that in the commands to determine the literal host names.

**When working with health alerts, you want to be in a LITERAL HOST name mindset only.**

Example commands to retrieve this information:

. /opt/CA/r11.3/autouser.DC3/autosys.ksh.DC3

autorep -M 1CCP\_DEV.virtual.net | awk '{ print $1 }' | awk -F "." '{ print $4 }' | sed '/^$/d' | perl -e 's/v/a/g' –p  
  
Example output from these commands :

cdpra02a0400

cdpra05a0400

cdpra04a0400

# Where are the script components located?

## 4.1 Location of the main wrapper script and it’s per environment control files.

All health alerts running on our servers are currently running out of my home dir.

/src/developers/u496456/home/health\_check

However you can find them checked into your source repository location at the following

/Projects/wellsfargo/ccp/ccp\_pub/health\_check

This means that your process accounts have been checking them out with each release

Here is an example from development server cdpra04a0400

Command : su - ccpetl1

Command : cd /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check

The main wrapper script is called

health\_check\_new

The main wrapper script per environment control file is called

health\_check\_new.<literal\_host\_name>.email.env

The main wrapper script control file is only responsible to set the command delimited email list .

> cat health\_check\_new.cdpra05a0400.email.env

FROM="Ab Initio Admin ID"

TO\_G\_14=louis.puskas@wellsfargo.com,Parikshit.Kamble@wellsfargo.com

**The main wrapper script’s environment files will be located in your process account’s home directory.   
They are not checked in.**

**You might see very old checked in versions in /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/\*env. Ignore Them.**

You will have to run these commands with your process account, if it’s the first time setting up health alerts.

mkdir -p ~/health\_check

cp /src/developers/u496456/home/health\_check/\*env ~/health\_check

You will see that you have copied over my latest script environment files now which can be updated without code promotion requests.

[cdpra04a0400] ccpetl1 /run/ccpetl1/dev/health\_check

> ls -rlt

total 19

-rw-r----- 1 ccpetl1 ccpdlog 185 Oct 24 09:29 check\_ai\_license\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 74 Oct 24 09:29 check\_daemon\_new.cdpra02a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 324 Oct 24 09:29 check\_daemon\_new.cdpra04a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 324 Oct 24 09:29 check\_daemon\_new.cdpra05a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 526 Oct 24 09:29 check\_df\_new.cdpra02a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 526 Oct 24 09:29 check\_df\_new.cdpra04a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 526 Oct 24 09:29 check\_df\_new.cdpra05a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 5 Oct 24 09:29 check\_dir\_perm\_new.cdpra02a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 5 Oct 24 09:29 check\_dir\_perm\_new.cdpra04a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 5 Oct 24 09:29 check\_dir\_perm\_new.cdpra05a0400.env

-rwxr-x--- 1 ccpetl1 ccpdlog 97 Oct 24 09:29 check\_ipcs\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 100 Oct 24 09:29 check\_mem\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 21 Oct 24 09:29 check\_new\_rpm\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 55 Oct 24 09:29 check\_per\_user\_process\_count\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 82 Oct 24 09:29 check\_top\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 6 Oct 24 09:29 check\_vmstat\_new.env

-rwxr-x--- 1 ccpetl1 ccpdlog 249 Oct 24 09:29 health\_check\_new.cdpra02a0400.email.env

-rwxr-x--- 1 ccpetl1 ccpdlog 249 Oct 24 09:29 health\_check\_new.cdpra04a0400.email.env

-rwxr-x--- 1 ccpetl1 ccpdlog 249 Oct 24 09:29 health\_check\_new.cdpra05a0400.email.env

## 4.2 Location of the individual check scripts and their per environment control files

The main wrapper script will call a series of support scripts in order built the full report.

They are located in the same folder as the wrapper script.

-r-xr-x--- 1 tlksecdm ccp\_src 1408 Oct 5 18:35 check\_daemon\_new

-r-xr-x--- 1 tlksecdm ccp\_src 3494 Oct 5 18:35 check\_df\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1830 Oct 5 18:35 check\_dir\_perm\_new

-r-xr-x--- 1 tlksecdm ccp\_src 2250 Oct 5 18:35 check\_EME\_status\_new

-r-xr-x--- 1 tlksecdm ccp\_src 6715 Oct 5 18:35 check\_ipcs\_new

-r-xr-x--- 1 tlksecdm ccp\_src 4290 Oct 5 18:35 check\_mem\_new

-r-xr-x--- 1 tlksecdm ccp\_src 4876 Oct 5 18:35 check\_netif\_speed\_volume\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1981 Oct 5 18:35 check\_new\_rpm\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1567 Oct 5 18:35 check\_per\_user\_process\_count\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1227 Oct 5 18:35 check\_semaphore

-r-xr-x--- 1 tlksecdm ccp\_src 6957 Oct 5 18:35 check\_top\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1831 Oct 5 18:35 check\_vmstat\_new

-r-xr-x--- 1 tlksecdm ccp\_src 1814 Oct 5 18:35 date\_less

-r-xr-x--- 1 tlksecdm ccp\_src 1891 Oct 5 18:35 parse\_date\_w\_date\_diff\_new

**The support scripts environment files will be located in your process account’s home directory.   
They are not checked in.**

**You might see very old checked in versions in /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/\*env. Ignore Them.**

# How does the script run ?

## 5.1 How to start the script

Very important : YOU MUST currently be logged into the Linux node you wish to start the script on!

Example if you want to start it on dev 04, you need be on dev 04.

[cdpra04a0400] ccpetl1 /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check

Yes, you have to do this step for each server you want monitored.

At the moment the health alerts are called with a nohup command.

Here is sample nohup command to start the health alerts on your servers.

Only pick one of these commands. If you’re trying to start the job on 04, pick the one that has nohup04.out in it.

nohup /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/start\_script.ksh > ~/health\_check/nohup02.out 2>&1&

nohup /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/start\_script.ksh > ~/health\_check/nohup03.out 2>&1&

nohup /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/start\_script.ksh > ~/health\_check/nohup04.out 2>&1&

nohup /src/ccpetl1/dev/abinitio/ccp/ccp\_pub/health\_check/start\_script.ksh > ~/health\_check/nohup05.out 2>&1&

## 5.2 How to check if the script is running properly

Check to see if you see the sleep process in the background :

export me=$(whoami);ps -alef | grep $me | grep 'sleep 580' | grep -v grep

Good Return :

0 S ccpetl1 105339 95209 0 94 14 - 25231 hrtime 09:48 pts/6 00:00:00 sleep 580

Bad Return :

You see nothing.

Check the timestamp of the health\_alert nohup file to see when it was last updated :

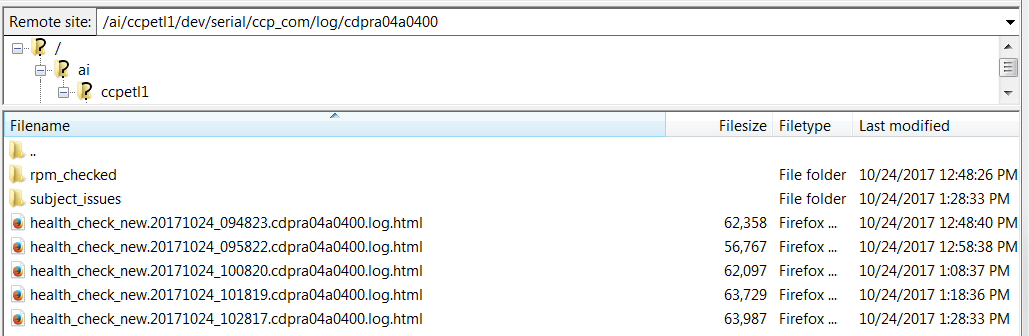
> ls -rlt ~/health\_check/\*nohup\*

-rw-rw---- 1 ccpetl1 ccpdlog 80 Oct 24 09:58 /run/ccpetl1/dev/health\_check/nohup04.out

# Where is the output located?

${CCP\_COM\_SERIAL\_LOG}/${HOST}

This location resolved for our example : /ai/ccpetl1/dev/serial/ccp\_com/log/cdpra04a0400  
  
You can go to this location with your desktop (s)ftp client and extract any report



# What items are checked by the Health Alerts ?

## 7.1.1 check\_daemon\_new

This script requires the following env file to run : check\_daemon\_new.<hostname>.env

The purpose of this check is to alert you if any of your critical ab initio support processes are not up and running before it becomes an issue.

You can see for the dev04 server these are the processes which should be up and running for our team:

> more check\_daemon\_new.cdpra04a0400.env

/usr/local/ab/abinitio-V3-3-2/bin/Xabworkloadd -log-dir /usr/local/ab/wlm

/usr/local/ab/ai-current-version/bin/XXabbridged /var/abinitio\_4/ab-bridge/ccpetl1-7076 -name ccp-ace-bridge-config-7076

/usr/local/ab/ai-current-version/bin/XXabbridged /var/abinitio\_4/ab-bridge/ccpetl1-7085 -name ccp-ace-bridge-config-corslab-7085

If one of them is not up and running you will receive a report like the following :



## 7.1.2 check\_df\_new

This script requires the following env file to run : check\_df\_new.<hostname>.env

The purpose of this check is to alert you if any of your critical ab initio san or nas storage locations are nearing full capacity before it becomes an issue.

The env file is used to select which filesystem locations you would like to monitor and when you would like to receive that alert. At what percentage you start to care about the capacity.

> more check\_df\_new.cdpra02a0400.env

/abinitio/dev|80

/abinitio/devemebackup|96

/abinitio/dev\_old|80

/ai/ccpetl1|80

/ai/ccpetl2|80

…..

If you go over 80% usage you will receive an alert.

You will not receive another alert until you’ve gone above 80%, or you’ve gone under 80% but then over 80% again.

Example going over the threshold : New Actionable Alerts are RED. Alerts which were reported but the % usage

hasn’t gone up are YELLOW.



## 7.1.3 check\_dir\_perm\_new

This script requires the following env file to run : check\_dir\_perm\_new.<hostname>.env

The purpose of this check is to alert you if a filesystem which should be open for world read/write access has for some reason had its permissions changed where the world cannot read or write to that location.

We don’t use this one heavily here, but we do check this one location.

> more check\_dir\_perm\_new.cdpra04a0400.env

/tmp

## 7.1.4 check\_EME\_status\_new

The purpose of this check is to alert you if the process account which is running your health alerts cannot access your Ab Initio EME (source repository) software for any reason.

The eme can be down or unreachable for a myriad of reasons. This check will report to you whatever the reason it is that it’s unreachable.

Attached is an example of when it reported the EME was unreachable due to an incorrectly executed storage format upgrade by the EDA support team.



## 7.1.5 check\_ipcs\_new

This script requires the following env file to run : check\_ipcs\_new.env

The purpose of this check is to alert you if you are running out of array or semaphore space on your servers.

In this example, you can see the server had used up 87.24% of the possible arrays on server cdpra04a0400



## 7.1.6 check\_mem\_new

This script requires the following env file to run : check\_mem\_new.env

The purpose of this check is to alert you if you are running out of all available memory on your Linux machine, before

it becomes a major issue. This alert will not alert you if you run out of physical memory start using swap. You would have to be close to running out of both before you received this alert.

In this example you can see two rouge java jar processes using up 38GB’s each of memory + a lot of development activity inside the GDE causing your memory issue.



## 7.1.7 check\_netif\_speed\_volume\_new

The purpose of this check is to alert you if one or more of your network interfaces has not been properly configured

by your system administrators.

While providing useful information about network setup on each of our servers, we have not encountered any issues

of incorrect network interface setups here by our admins to embed here.

## 7.1.8 check\_new\_rpm\_new

This script requires the following env file to run : check\_new\_rpm\_new.env

Inside you will find a setting for how far back each scan should look to find new packages.

> more check\_new\_rpm\_new.env

DAYS\_OLD\_THRESHOLD=3

The purpose of this check is to alert you if your system administrators have been installing new packages on your

server outside of a green zone time when this is supposed to happen.

We see the system administrators regularly altering our environments here without notice. This should not be

allowed, but here it’s ok. Here is an attached example from today.



Packages which have already been announced will not be re-announced in health alerts.

## 7.1.9 check\_per\_user\_process\_count\_new

This script requires the following env file to run : check\_per\_user\_process\_count\_new.env

Inside you will find a setting for when a report should be sent, and one to control which users should show up in the

report based on a minimum number of processes to become relevant to the report.

The purpose of this report is alert if you one user on a machine has spawned too many processes.

> more check\_per\_user\_process\_count\_new.env

PROCESS\_COUNT\_THRESHOLD=1000

PROCESS\_SHOW\_THRESHOLD=30

From this attached example from today. You can see the alert was sent because the user has more than 1000 active

processes on the server.



## 7.2.1 check\_top\_new

This script requires the following env file to run : check\_top\_new.env

The purpose of this report is alert is to look for suspicious output from the top utility built into Linux.

This check looks for the following :

1. A load average on the system which is much higher than average for that particular for that server.

The settings for this report are set on this line : LOAD\_THRESHOLD=150



1. A long running process which has taken a single cpu hostage for hours without freeing it up. All while running at a very high cpu percentage.

The settings for this report are set on these two lines : CPU\_THRESHOLD=90

TIME\_THRESHOLD=1500

So this would report any process which has been running on one cpu for longer than 1500 cpu seconds and is still using 90% or more of that cpu’s capacity.



1. An i/o wait cpu usage percentage higher than that you have specifcied is a bad or high value.

The settings for this report are set on this line : IO\_THRESHOLD=5



## 7.2.2 check\_vmstat\_new

This script requires the following env file to run : check\_vmstat\_new.env

The purpose of this check is to check your vmstat output for evidence of thrashing on your servers.

<https://en.wikipedia.org/wiki/Thrashing_(computer_science)>

You can see in the env file a specific setting for what you have decided is a bad major page out value.

> more check\_vmstat\_new.env

PO=10

