

CONTROL-M - JOB FAILURE RERUN COUNT...



Edison Santos 119 posts since Oct 20, 2014

CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 5:30 AM

Hi guys,

Here I'm with a new question.

Reading CONTROL-M administrator manual for version 9, I found that the "Job's number of failures" exists for CONTROL-M, but, its value is not loaded to no one variable (Job's number of failures.....Variable Name: None).

The value of consecutive errors is stored by CONTROL-M and is used, for example, for the user to determine when to stop a cyclic execution by a certain number os RERUNS.

This information also goes to the CTM JOB LOG with code 5100 (RUNCNT). I've also read something about variable %%RN (HERE: <https://communities.bmc.com/thread/151819?start=0&tstart=0>), but, it looks like an exclusive MAINFRAME option when integrated with other BMC tools.

My purpose is to create tickets automatically to every first CTM JOB failure into a non BMC ticket solution.

I was thinking to use the "SendAlarmToScript" option only if the "NOT OK RUN COUNT" by the JOB should be equal to 1 (this count should be reset after a first ok next run.

I was thinking to extract this information from the JOB LOG. but. this "alert" will be running on an EM server, while ctmlog must be executed into the CTM Server only.

If I can't solve this problem with "SendAlarmToScript", I should use "Do Actions / Jopb's Number of Failures / =1" sending a message for a SCRIPT.

I also have a doubt about the parameter "HandleAlertsOnRerun": Is it reset at each run ok?

Could anybody give some help?

Best regards,

ES

Tags: [control-m administration](#), [failure count variable](#), [send alarm to script](#)

CONTROL-M - JOB FAILURE RERUN COUNT...



Adriano Gomes

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 6:27 AM

Hi [Edison Santos](#)

I am very glad to have You here [Control-M](#) Community!

Here you have some clarification:

1) **"HandleAlertsOnRerun"**: Is it reset at each run ok?

No. This CTMEM System parameters will instructs GTW process to "handle" alerts issued to CTMEM Alert windows for a given job automatically when the operator performs job rerun, regardless this new run status.

Regarding the main question :

I would approach this use case by configuring CTMEM systems parameters to run a script/program on each alert received. The reason is: CTMEM call the scripts/program and provide the Job current RUNCOUNT as part of alert message fields as an argument variable and by writing an integration script you can parse variable arguments as you wish.

The CTMEM **"SendAlarmToScript"** along with some other ctmem system parameters configuration will make CTMEM GTW to perform such integration script execution at CTMEM BOX and you can leverage CTM A API to get job log by calling "`ctm run job:log::get <jobId>`" service after performing authentication and retrieve whatever info available on JOB Log.

CTM A API is very useful and can perform most of GUI operations over command line local to em or remotely. It is available as of 9.00.400 or later.

here you can have further details on how to activate CTMEM Integration and **"SendAlarmToScript"** configuration.

[Integrating Control-M with Ticketing and Monitoring Systems](#)

My Best Regards

A>Gomes



[Edison Santos](#) 119 posts since Oct 20, 2014

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 9:41 AM

in response to [Adriano Gomes](#)

Hi Mr. Adriano, It is a pleasure for me to meet with you again! Thanks for your attention!

CONTROL-M - JOB FAILURE RERUN COUNT...

About the parameter "HandleAlertsOnRerun", my bad! I know what this parameter means and I just did a wrong copy/paste of the parameter name. The correct parameter name for the question is "IdenticalXAlertHandling".

I'm still reading your article and it seems to be the same solution that I was thinking of implement, but I was worry about duplicating the tickets by successive executions of the same ABEND (for a cyclic JOBS, for example).


But, I think that the default value of "IdenticalXAlertHandling", prevent this, because of successive alerts will be blocked to SNMP/SCRIPTS. This, if next non-consecutive NOT OK alerts will not be treated as an "identical alert", and that is my first question (which was with the wrong parameter name).

Regards,
ES



Adriano Gomes

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 10:54 AM

 in response to [Edison Santos](#)
Hi [Edison Santos](#)

My pleasure! Count on me!

first of all, IdenticalXAlertHandling is not related to Alerts issued by Job/Shout mechanism, but related to CMS self maintenance monitoring system that proactively monitors several availability scenarios and issue XAlert Message to CCM only GUI.

Now, regarding integrating CTMEM with any Ticketing System (Not Remedy - supported "natively") with messages related with Jobs behavior, indeed you need to decide what method you are going to apply in order to avoid multiple tickets for the same job instance/job run count. That is mainly because the "Alert" have its own lifecycle, so for a single Job Alert Message type (I) you can have multiple messages type (U).

You can write your scripts/programs to perform like this :

1. Open the ticket for type (I) only, also you can filter on severity and message fields.
2. retrieve ticket Number/status and store on ALARM table or some local file with <control-m.alert_id.alert_date.orderid.runcount>.dat, that will help you to avoid duplicating due to each newer alert type (U) will have original date, orderid and run count unique for all changes during lifecycle.
3. Update the ticket status when operator change Alert status(excluding Handled, Notes), severity ;
4. Now for Closing the tickets the you can have at least two scenarios :

1) the next job run finished OK - The job will not issue any new Alert, but it is not the same RUNCOUNT instance anymore.

CONTROL-M - JOB FAILURE RERUN COUNT...

You can have either the **"HandleAlertsOnRerun" system parameter activated and** a new alert type (U) for the current failed job instance will be automatically generated that can be used to trigger the ticket closure, **or manually**, the operator will change the alert status to "Handled" in order to trigger ticket closure.

It is very important to notice that Ticket Systems does have two distinct status that is used to represent ticket end of lifecycle, like **"RESOLVED"** and **"CLOSED"**, so you can move the ticket to "Resolved" automatically with **"HandleAlertsOnRerun" and perform "CLOSED" by adding a "NOTE" to Alert after the Operator verify the Next job Instance is finished OK (even after the alert is in Handled status).**

2) the next job run finished NOK - The job will issue new Alert (same order id, new run count), the same Job, but it is not the same RUNCOUNT instance anymore.

This use case, if you want keep the same ticket opened for the previous job run id, you can always list ALARM table or the local file with <control-m.alert_id.alert_date.orderid.runcount>.dat and look for previous runcount ticket number/status "NOT CLOSED" and add new incident task or log information.

You can also have a detached process that monitors ALARM table or the local file with <control-m.alert_id.alert_date.orderid.runcount>.dat files for incident status tickets that is still "NOT CLOSED" and use CTM A API to get the job details to validate the job status and close the tickets automatically.

Hope it helps you get more doubts

My Best

A>Gomes



[Edison Santos](#) 119 posts since Oct 20, 2014

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 11:38 AM

 in response to [Adriano Gomes](#)

Adriano, thanks a lot for your help! You are really helping to clarify this theme!!


But, like your expectations, your explanation is moving me for more doubts.....:)

The RUN COUNT number is not incremented for NOT OK RUNS???



[Adriano Gomes](#)

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 27, 2019 12:12 PM

 in response to [Edison Santos](#)

Hi Edison

Yes, that is very tricky!

CONTROL-M - JOB FAILURE RERUN COUNT...

think of that the orderid and run_count that opens the first ticket, when the job is rerun for verify resolution, it is not the same run count anymore.

This new run_count can finish OK/NOK, so new alerts can be issued to Alert screen on both cases, that is not the same run_count that has opened the ticket.

So you have to write your code based on what approach you want to follow, be that:

- 1) A single ticket to ALL failed job runs for the same order_id
- 2) A new Ticket on each failed RUN_COUNT job instance on the same order_id.

No matter what, you will have to decide how the process will be when Alerts is being manipulated by operators so you can code appropriately.


My Best

A>Gomes



[Edison Santos](#) 119 posts since Oct 20, 2014

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 28, 2019 4:14 AM

 in response to [Adriano Gomes](#)

Hi Adriano, thanks again for your attention and support!!

My point is:

- 3) To open a ticket for the first NOT OK after one subsequent OK.

What I mean is, for a cyclic job, for example, when first NOT OK occurs after an immediately OK execution, the ticket will be opened. For the subsequent NOT OK, no, until it has an immediately previous execution OK, and so on.

It can't open anyone ticket if the NOT OK run doesn't have an immediately previous execution OK.

With a RUN COUNT FAILURE variable, I could create a ticket for every COUNT FAILURE = 1, and I wouldn't need to store any kind of data on the CTM/EM side (at this moment).

illustrating:

For each run of a specific ORDERID (cyclic, for example), it must do:

1. OK - NONE
2. OK - NONE
3. NOT OK - Open a ticket
4. NOT OK - NONE
5. NOT OK - NONE

CONTROL-M - JOB FAILURE RERUN COUNT...

6. OK - NONE (in the future, maybe it will be integrated with a CLOSE TICKET action)
7. OK - NONE
8. NOT OK - Open a ticket
9. NOT OK - NONE
10. NOT OK - NONE
11. OK - NONE (in the future, maybe it will be integrated with a CLOSE TICKET action)
12. OK - NONE.....and so on!!


Regards,

ES



[Adriano Gomes](#)

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 28, 2019 6:19 AM

 in response to [Edison Santos](#)

Hi Edson

For cyclic jobs you can stop cycling when NOT OK happens as an option and then open a ticket for investigate the issue until you have the resolution to solve and close it, but it means to me that your use case is pretty much like until you have an opened ticket for an orderid do nothing (I mean, you can just update Alarm NOTES with the existing Incident ticket number or the ALARM.ticket_number table field row for that ALERTID).

Like I have mentioned, you must keep at ALARM table or at local control file, the ticket status for such ORDER_ID regardless it RUN_COUNT, so this way, for every alert type (I) you must read and check whether ORDER_ID have an opened ticket already or not yet.

Also, it is very important to notice that some issues reported as incidents can age and take long path until cleared and definitely solved, and sometimes the root cause is not the same for each job runcount failed abend code reason. I would recommend You have a talk with Service Manager or the Incident process owner in order to understand what is the practice for such use case, and lastly, some ITSSM tools can do ticketing association automatically, so the first ticket will hold all the remaining related tickets reported for problem analysis. This way, you could open a ticket for each distinct runcount as usual and the Ticketing tool would automatically relate then one to another.

I hope that it helps enlighten your path.

My Best


A>Gomes



[Edison Santos](#) 119 posts since Oct 20, 2014

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Nov 28, 2019 10:57 AM

CONTROL-M - JOB FAILURE RERUN COUNT...

 in response to [Adriano Gomes](#)

Hi Adriano, thanks again, and again, and..... ..

I'm almost there..... thanks for your patience and attention ()!

I know that I could use a STOP CYCLIC to prevent loop runs of my cyclic job. That is not my point, it was just using it as an example for illustration.

About your recommendation, I think I shouldn't open tickets for each ORDERID, because the NOT OK run can be a second try to run a problematic job, and, for this case, the ticket is already opened. In my environment, I have a lot of combinations of retries em jobs that are "reused", and could happen to have another NOT OK run that must have a new ticket (for the same ORDERID).

You are completely right about the theme of incidents, and I know a little bit about it. For now, I'm focused on the technical solution. The specifications here say that I must not open tickets for a consecutive NOT OK run and always for one that it is preceded by an OK run.

I have all on my mind to construct this solution and your document will help me with much more less stress on my mind, especially the detail about the RUN COUNT (I'm thankful for this), but, the question here is that I was trying to find some shortcut that could make me use much less processing and less code (the FAIL RUN COUNT, if it existed (calm, I already know that it doesn't) would do exactly this for me).

Could you talk a little bit more about the "alert type", or recommend some friendly text that explains about it? Its generated for each NOT OK alert or just once for an ORDERID?


Regards,

ES



[Adriano Gomes](#)

Re: CONTROL-M - JOB FAILURE RERUN COUNT VARIABLE Dec 2, 2019 10:52 AM

 in response to [Edison Santos](#)

Hi Edison

Never seen any lecture related to CTM integration with Alert mechanism.

There are two possible values for an **alert_type**: "I" and "U" and they are tied to the same **alert_id**.

So, for an unique **alert_id** you can have a **single** issue **type (I)** and **multiple** issues **type (U)** with the same **alert_id**. Each Alert itself carry the same order_id and run_counter all the way until the Alert is handled and finished. That is the logic.

Alert_type is not related with Job status but with the **Alert lifecycle itself**. If the Alert was issued due to a failed active Job, then the **order_id** and **run_counter** will be part of alert fields used to invoke the integration

CONTROL-M - JOB FAILURE RERUN COUNT...

script. There are Alerts issued to EM with messages not related to jobs that do either do not have **order_id** or the **order_id** is coded for internal specific maintenance purposes.

Thanks for sharing your requirements "The specifications here say that I **must not open tickets for a consecutive NOT OK run and always for one that it is preceded by an OK run.**"

So, If you want each and every failed run to be fully solved and have an associated incident ticket to that specific pair of orderid/runcount (the failed run), like you just mentioned, "because the NOT OK run can be a second try to run a problematic job," remember that while recovering the past job execution the **ORDER_ID is on its next run counter** (for a cyclic that do not stop when the previous ran fail or for a regular job itself) and the alert_id for the next message will also change.

As each ALARM event **alert_id type (I)** issued to the EM Alert Window will be associated to one UNIQUE specific **order_id, run_counter** and naturally for a failed run that will trigger the incident ticket creation, **you must to store the incident ticket number opened in a place where you can query/read back for status.**

Please notice that if the next run for a cyclic/regular job run fail on its rerun, there will be a **new** ALARM event **alert_id type (I)** issued to the EM Alert Window for the same Order_id but different **run_counter**. What I am trying to say is that the process behind the Alert Window can support ticketing integration but the run_counter used to solve the failed run associated with the incident ticket will never be the same, thus the script must query ALARM table for past failed alerts for that (order_id and run_counter - 1) and verify whether there is already a ticket opened or not.

On our shop, we first query current job run status. As I mentioned, the Alert is an Alert and do not carry job status, just a message that is defined for that job situation.

My Best

A>Gomes