We have all been there. You create a job with a tMap and it fails due to some problem in the tMap config. Many times it is due to something connected to data being null. Maybe you have a "compareTo" checking a String that is null or a "parseInt" trying to convert some non-numeric text to an Integer. They are common issues, but they are generally pretty tricky to identify so that you can cater for situations in which they occur. That is if you do not use a trick that is very useful and surprisingly very rarely used. Talend actually provide a mechanism to help you pinpoint these issues, but for some reason they do not publicise it well. This post will show you how to use it.

When you use a tMap component, by default it is set to die on error. So if there is an errror with processing data, the job will die and you will receive a Java runtime error (like a NullPointerException). However, this does not pinpoint where the error occurred (without digging through the error stack) and certainly doesn't tell you the data that caused the error. However, this does not need to be case. You can set the tMap to not die on error and to actually have an output specifically for errors. The screenshot below shows how to enable this....

In the top left corner (where the green box is) there is a button. Press this and it will reveal the Property Settings window. Untick the "Die on error" tick box. This will reveal a new output for the tMap with pre-configured columns for "errorMessage" and "errorStackTrace". Now whenever there is a Java runtime error caused by the tMap, you will get an error that will not kill the job.

But on its own, this isn't much help. To add value to this, you will need to add all of the columns that you are processing to the error output. Then when an error occurs, you will log the datarow that caused it. An example of this can be seen below....

The columns boxed in red are generated automatically for you. The columns boxed in green have been added by me as an example of how to return the datarow that has errored.

The next thing to do is to return this data to somewhere useful. Since this sort of error trapping will likely take place during development, it probably makes sense to dump this to a tLogRow component. However, this could be sent to a file or a database. For this quick tip, I have chosen to dump the data to a tLogRow component. When there is an error, we can see exactly what caused it. The error below was caused by trying to convert the "newColumn2" column from a null to a number....

```
Starting job Test at 19:07 16/08/2015.

[statistics] connecting to socket on port 3574
[statistics] connected


 #1. tLogRow_2
+----------------+-------------------------------------------------------------------------------------------------------------------------------------+
| key            | value                                                                                                                               |
+----------------+-------------------------------------------------------------------------------------------------------------------------------------+
| newColumn      | newColumn = test                                                                                                                    |
| newColumn1     | newColumn1 = tester                                                                                                                 |
| newColumn2     | newColumn2 = null                                                                                                                   |
| errorMessage   | null                                                                                                                                |
| errorStackTrace | java.lang.NumberFormatException: null
    at java.lang.Integer.parseInt(Unknown Source)
    at java.lang.Integer.parseInt(Unknown Source)
    at local_project.test_0_1.Test.tFixedFlowInput_1Process(Test.java:1149)
```

```
    at local_project.test_0_1.Test.runJobInTOS(Test.java:1721)
    at local_project.test_0_1.Test.main(Test.java:1578)
 |
+-----------------+-----------------------------------------------------------------------------------------------------------------------------------------+
[statistics] disconnected
Job Test ended at 19:07 16/08/2015. [exit code=0]
```