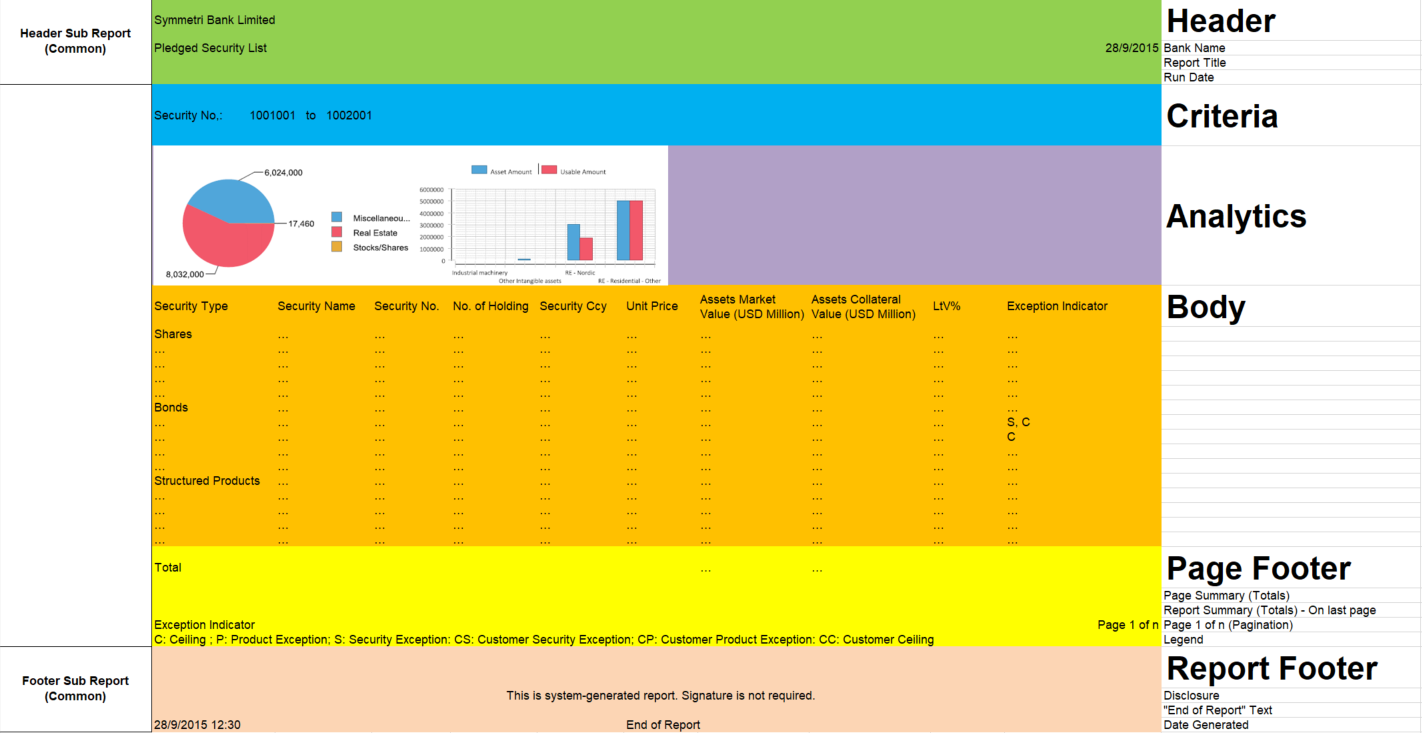
**Report Layout**



**Reporting Criteria**

1. If report filter is range show as "<range\_start> to <range end>"  
   Example:  
   Transaction Date: 1-Jan-2015 to 30-Jun-2015
2. If there is no criteria in report filter field, show as "All"  
   Example:  
   Branch: All
3. If report filter is a list show as "<range\_start>…<range end>"  
   Example:  
   Ayala Branch  
   Dela Costa Branch  
   Legaspi Branch  
   Paseo Branch  
   Zuelig Branch  
     
   Example:  
   Branch: Ayala Branch, Dela Costa...

When report query does not produce resultset, report will still be generated even without data. This serves as indication that report has not been inadvertently skipped.

For End/Start of Day reports, the report format (PDF, HTML, etc) is defined in Registry (Default Batch Report Format).

Styles must be driven by style template.

**Report Filter UI Components**

These are the components to be used in Report Parameter Form.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Text Filter | List Filter | Multi LOV Filter |

Report Filter Behaviour and Operation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| textfilter | Allows user to input filter value and operator  Equal  Not equal  Future:  Contains  Starts with  Ends with |  |  | $X{EQUAL, <column\_name>, <parameter\_name>}  $X{NOTEQUAL, <column\_name>, <parameter\_name>} |
| numberfilter | Allows user to input numeric filter value and operator  Equal  Not equal  Greater than  Greater than or equal to  Less than  Less than or equal to | **valuerange** component can partially cover the same features.  For standardization for report parameters, refrain from using this. |  | $X{EQUAL, <column\_name>, <parameter\_name>}  $X{NOTEQUAL, <column\_name>, <parameter\_name>}  Greater than  $X{GREATER, <column\_name>, <parameter\_name>}  Greater than or equal to  $X{[GREATER, <column\_name>, <parameter\_name>}  Less than  $X{LESS, <column\_name>, <parameter\_name>}  Less than or equal to  $X{LESS], <column\_name>, <parameter\_name>} |
| daterange | Allows user to input From-To dates.  If both dates are provided, condition will be treated as between From and To dates.  If From Date is not inputted, condition will be treated as Less Than or Equal to To Date.  If To Date is not inputted, condition will be treated as Greater Than or Equal to From Date.  If both dates are not provided, no date filter will be applied. |  | Examples:   1. Transaction Date 2. Maturity Date | $X{[BETWEEN], <column\_name>, <left\_parameter\_name>, <right\_parameter\_name>} |
| listfilter | Allows user to pick filter conditions based on predefined values. | Use only if predefined values are limited regardless whether it’s deliverable data, from common Reference Data table or from dedicated Reference Data table.  Roughly 100 or less predefined values should be a good range. | Examples:   1. Account Status 2. Currency | $X{IN, <column\_name>, <parameter\_name>} |
| valuerange | Allows user to input 2 values for the range.  If both values are provided, condition will be treated as between.  If first value is not inputted, condition will be treated as Less Than or Equal to the second value.  If second value is not inputted, condition will be treated as Greater Than or Equal to first value.  If both values are not provided, no filter will be applied. |  |  | $X{[BETWEEN], <column\_name>, <left\_parameter\_name>, <right\_parameter\_name>} |
| numberrange | Allows user to input 2 numeric values for the range.  If both numeric values are provided, condition will be treated as between.  If first numeric value is not inputted, condition will be treated as Less Than or Equal to the second numeric value.  If second numeric value is not inputted, condition will be treated as Greater Than or Equal to first numeric value.  If both numeric values are not provided, no filter will be applied. |  |  | $X{[BETWEEN], <column\_name>, <left\_parameter\_name>, <right\_parameter\_name>} |
| checkboxfilter | 2 checkboxes with True and False values | Use this if the column being filtered is of boolean type. |  | $X{EQUAL, <column\_name>, <parameter\_name>} |
| checkboxynfilter | 2 checkboxes with Y and N values | Use this if the column being filtered has value of Y or N. |  | $X{EQUAL, <column\_name>, <parameter\_name>} |
| multilovfilter | Allows user to pick filter conditions based on predefined values. | Use if predefined values are coming from transactional data source that continuously grow in count.  It is easier to choose the values from LOV rather than scrolling over huge rows of list (as offered by listfilter).  Nonetheless, listfilter has also advantage over multilovfilter of the options are much less since user can just quickly click the options that must be included in the filter. |  | $X{IN, <column\_name>, <parameter\_name>} |
| textinput | Allows user to input value that can be used inside the report | Use this if the field value to be entered is not for filtering but input needed to generate the report. | Example: |  |
| date | Allows user to input exact value for date | Use this if the field value to be entered is not for filtering but input needed to generate the report. | Example:  Deposit Account Interest Projection Report requires date which will be used as basis for calculation. |  |
| datetime | Allows user to input value that can be used inside the report |  |  |  |
| integerinput | Allows user to input value that can be used inside the report | Use this if the field value to be entered is not for filtering but input needed to generate the report. |  |  |
| listofvalues | Allows user to input value that can be used inside the report | Use this if the field value to be entered is not for filtering but input needed to generate the report. |  |  |
| combobox | Allows user to input value that can be used inside the report | Use this if the field value to be entered is not for filtering but input needed to generate the report. | Example:  Report Format to generate – PDF, CSV, HTML |  |

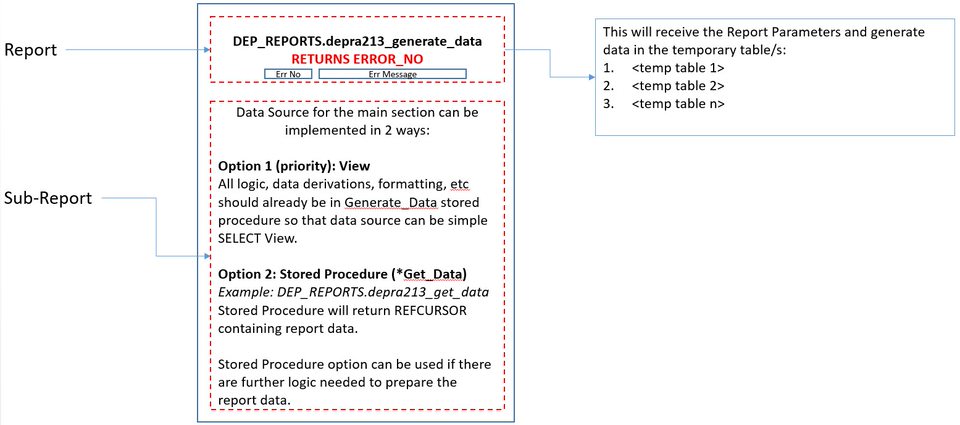
**Implementation for Usual Parameters**

|  |  |
| --- | --- |
| Parameter | Parameter UI component |
| Branch | multilovfilter |
| Account Type  Loan Type/Loan Sub Type | multilovfilter |
| Currency | listfilter |
| Borrower/Client No | multilovfilter |
| Loan/Account No | multilovfilter |
| Date range | daterange |

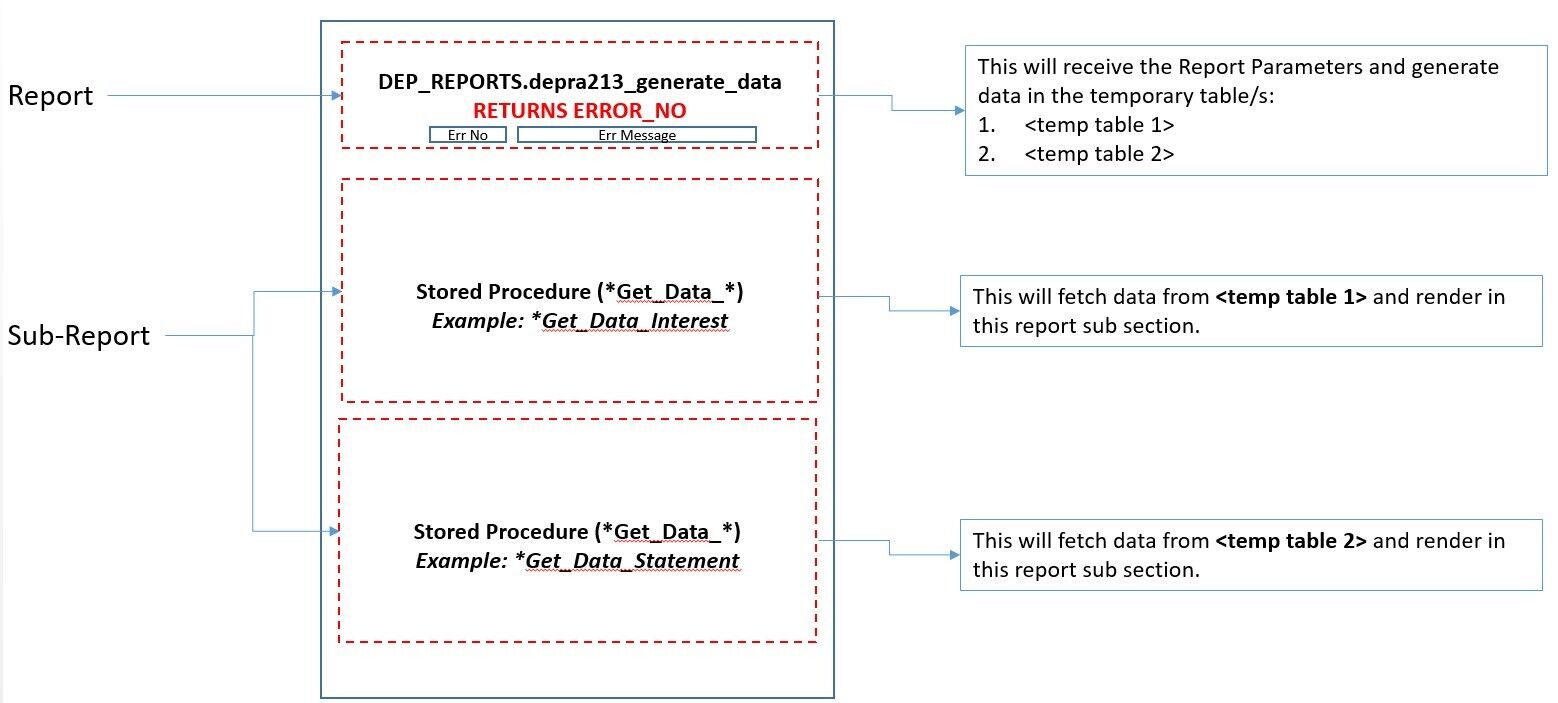
**Reports with Stored Procedure and Error Handling**

There are cases where report data is not readily available in the database and needs stored procedure to generate report data.

* Use Global Temporary Table for automatic housekeeping. If regular table is used, make sure housekeeping of old data is deleted first prior to generation of new report data.
* The main report data source should be stored procedure and will be responsible to generate the report data into temporary table/s.
  + Naming convention: \*GENERATE\_DATA
  + Returns Error\_No - There should be error handling in the report data generation logic. Logging should also be implemented (Debug, Error, etc) for easy investigation.
* There could be several sub-reports depending on what data to be shown for each section. Each sub-report should have its own temporary table.
  + The sub-report can be simple SELECT table or view where one of the underlying table is the temporary table where the report data is stored by \*GENERATE\_DATA stored procedure.
  + Data source can also be another stored procedure. Naming Convention: \*GET\_DATA\*. The stored procedure will only fetch the generated data from the temporary tables.

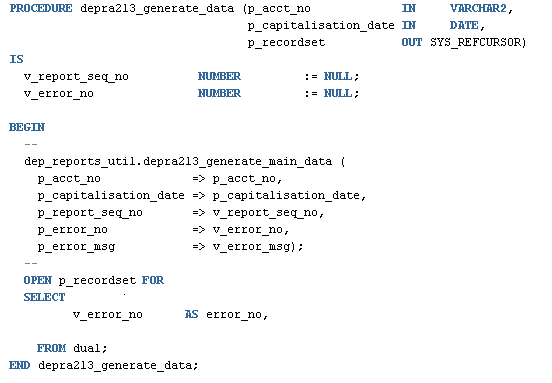


If there will be multiple sub-reports, each sub-report will have its own \*GET\_DATA\* stored procedure.

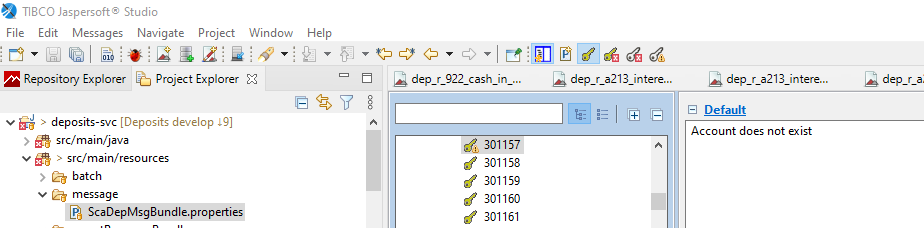


**Important things to prepare…**

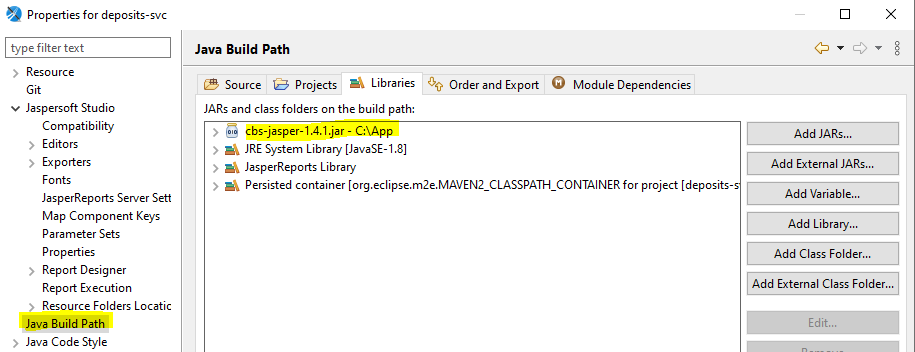
* The main Stored Procedure will be the one to handle the error during generation, hence, its OUT parameter is REFCURSOR containing Error\_No.



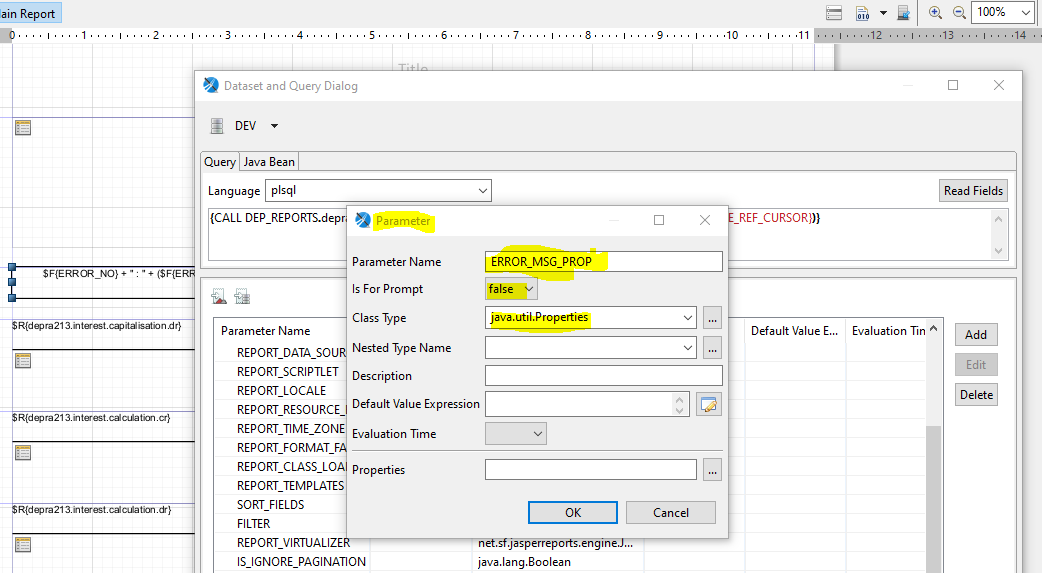
Error code must be included in the \*MsgBundle.properties in java. This is where the Error Message will be retrieved from.



* In you Jasper “Java Build Path” > Libraries cbs-jasper-1.4.1.jar must be added as external jars, this serves the linked between the return error code and the \*MsgBundle.properties to get equivalent error\_code from the managed resource bundle.



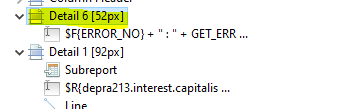
* In JasperReport make sure that **ERROR\_MSG\_PROP** parameter is defined, **Is for Prompt = false**, **Class Type = java.util.Properties**.

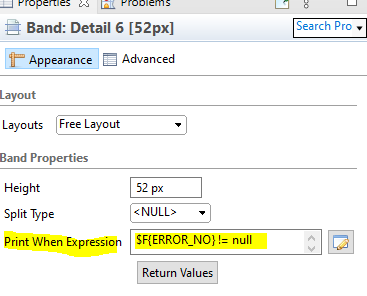


**Display Error Code and Error Message  in Reports**

**Step 1**

Add Band Details for Error messages and define Print When Expression “$F{ERROR\_NO} != null” this will ensure that this band detail will display when there is an ERROR\_NO.

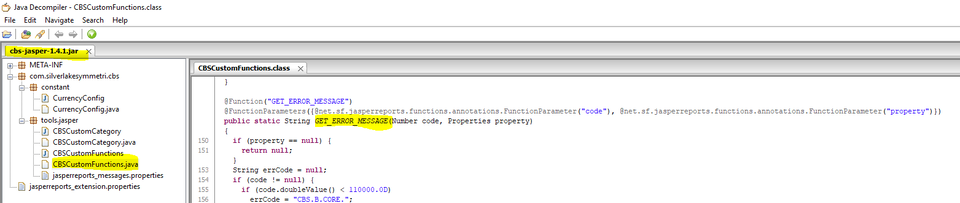




**Step 2**

To display the error\_code and error\_message using the \*MsgBundles.properties, Add “Text Field” with expression **$F{ERROR\_NO} + " : " + *GET\_ERROR\_MESSAGE*($F{ERROR\_NO}, $P{ERROR\_MSG\_PROP} )**

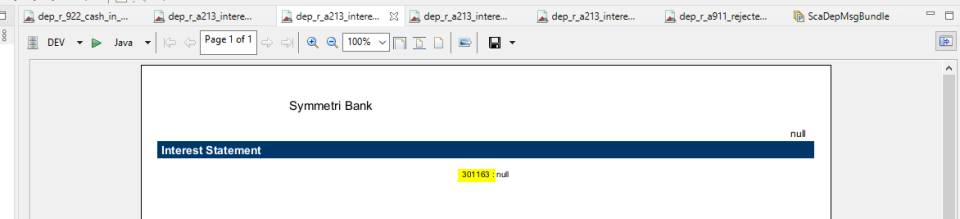
***GET\_ERROR\_MESSAGE***is a custom function defined in cbs-jasper-1.4.1.jar that serves the linked to java \*MsgBundles.properties.



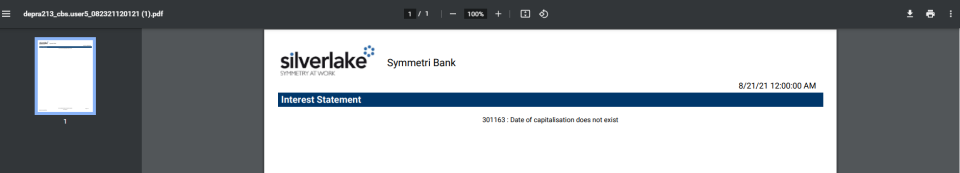
**Step 3 …. Testing**

Note that we can only test this when deployed in weblogic and through our application CB9.  If we will test the ERROR\_MESSAGE in JasperSoft it will return NULL. See below screenshot…

**JASPERSOFT OUTPUT**

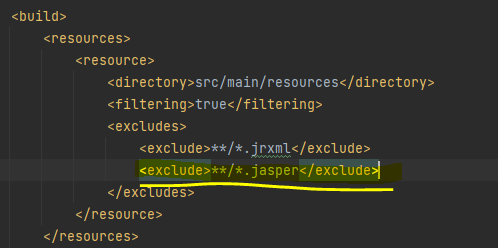


**FROM WEBAPP OUTPUT**

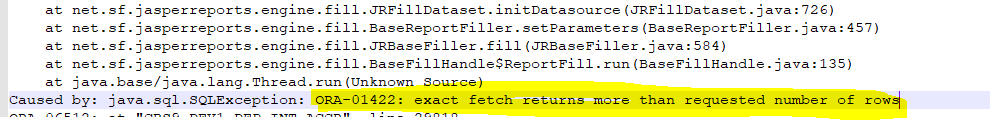


**IMPORTANT NOTES:**

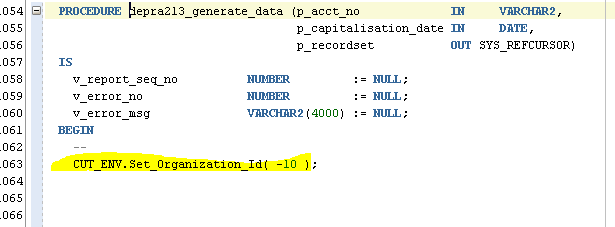
* When testing in local environment via weblogic, add this in your pom.xml and remove after the test. If this is not defined reports generated will failed or error.



* For reports with stored procedure it is normally encountered this error, this is due to the CSD\_SYSTEM\_V or it fetch all the ORG\_ID. Viewing reports via JasperSoft, it can’t identify which user organization to use that lead to this error.



* To resolve this we need to define the ORG\_ID during test, and should be removed after the development/testing of reports.



* Additionally to generate logs for investigation, set the logger level to DEBUG before calling cut\_logger.init. These lines should be removed after the development/testing of reports. ​​​​​​​

