Description	Component	Summary	Test Scenario
Cannot export highlights to Excel from estimation	Rules Studio	In SAS macro utl_export_excel.sas, we get a list of column names from SQL view highruleview. If the client specified the entity key column (typically TST_UNQ_TRADE_ID) in config property highlight_results_columns, this view has two columns with exactly the same name (both named TST_UNQ_TRADE_ID). Surprisingly, this works OK except in the case of getting the list of olumns(which now has that column name twice). When the list is used later to manipulate columns, the presence of the duplicate causes a syntax error.	Run an estimation on a rule that produces highlights and export the highlights to Excel.
Able to promote rule from testing to production which has unsatisfied dependencies	Rules Studio	Create an ARD rule and promote it to testing. Create a queue rule that uses %hasRuleFired to check the ARD rule and then promote to production. The promotion of the queue rule to production should fail. However, it succeeds and later gives a null pointer exception when clicking the "current" folder under deployments.	When promoting rules with dependencies from testing to production, the user will receive an error message instead of success
Error Entering BBW Criteria with Merchant SIC	STRATEGIES	When specifying BBW merchant SIC code = 6011, receive the 'unexpected error' screen. Error displays while scrolling down to the save button but prior to being able to click on it. Other BBW criteria can be specified successfully. As a result, unable to set/save a queueu with a Merchant SIC code	Entering an invalid SIC code will not produce the generic error page, instead a message box will display stating "The following codes are invalid: ####" Entering a valid code and tabbing off the field will save the code.
backport / release java Client library that reads as well as writes transactions	BPI	Capability to read segments as well as write them, Java Client Library - turn object into a byte array and reverse.	Backported Uts and read in a Latin transaction set, compared the RRR values from another SAS job reading the same transaction set. Confirmed all values matched.
write do not stream SAS code to the SAS workspace server during rule test and estimation	Rules Studio	Webapp streams generated rules file to SAS ION for rule testing and estimation. When the file is really big, it appears to make SAS IOM hang and the SAS job never starts. The fix is to create the generated rules file in an area accessible to both webapp and SAS and submit a tiny job to %INCLUDE it.	Run any rule test or estimation and examine the resulting SAS log to assure that the generated rule code was %INCLUDE'd rather than appearing inline. Customers who had issues starting estimation jobs should notice that the estimation jobs and rule tests will start up now as opposed to hanging during startup.
Recovery jobs fail for VXX tables after applying hotfix 11	Batch jobs	Found in macro etl_trans_vxx_loader.sas - a where clause was to be replaced by new logic however old line was not deleted and resulted in 2 "where" clauses being submitted and generated an error.	Contact Fraud Management if more information is required
Unable to Apply Transaction Block for Save Pending Result	UI	Unable to apply a transaction block code only when selecting save pending, even though the queue was configured to allow a transaction block. Currently only checks the SYSTEM BLOCK code when determining whether to use the seetting of "with block" or "without block". Therefore when only a transaction block is applied, system block is empty, the "without block" is selected. As a result, transaction block is rejected. FIX: Change the code to check both system block and transaction block selections.	Precondition: The alert queue is setup with save pending with both system block codes and transaction block codes. Login SFM, servicing an alert from the queue. Select Call Result "Save Pending" Select a transaction block, then date Save the call result. Expected result: Alert action is saved, ,and appears in the activity table when viewing the call result.
client library should return a transaction given a byte array	API	API not parsing out transactions as desired	Additional API created
ETL for frh_user_variable table (4019) has potential error	ETL	Issue occurs when there is a segment that has two or more keys. When a user variable rule that uses a variable in a segment keyed by two or more keys is deployed, there is an entry in the fsx_user_variable for each key. An attempt to insert remaining rows, results in an error as the frh_user_variable_dim RH table has a unique index based on build_id and uservar_field_name	Trigger will ignore multiple keys and only put one row in the table
When updating db2.9.5 to fix pack 10 it causes an sql passthrough query to fail	Batch Jobs	Queries in recovery jobs that have comparisons based on a timestamp column receive an error "Unexpected token was found"	timestamp literal is converted to an internal timestamp value. Rerun of recover and gap filler jobs performed.
BPI changesto prevent the RFC1323 zero window problem	BPI	Steadily decreasing throughput within a single BPI file is observed. The problem occurred due to the window scaling algorithm in TCP when the RFC1323 feature is enabled. RFC1323 was intended for transmission of very large files or for use in communication netowrks. The BPI sends transactions without ever waiting for a response and this causes the TCP buffers to fill up.	Contact Fraud Management if more information is required
Provide accurate reporting when running the 3005 job - bulk alert status	Batch jobs	When running build alert status job, not receiving feedback if the data in the input file contains inconsistent information such as no corresponding records in the fcm_alert table or no messages for failed updates. These records are not written to the rejects file.	This was tested by editing a bulkAlertStatus.csv file which contains errors for use in validation. Also, valid alerts were added to status but no matching alerts in the fcm_alert table.
Recovery jobs may miss a small number of records due to rounding of lower bound value	Batch jobs	When setting the lower/upper timestamp boundaries for copying data from the SOR, the logic needs to account for MODE=ALL, MAXRECS=# and neither of these job parameters. If not set correclty for the case being executed a few records may be skipped in the copying process due to using integer values derived from timestamp values.	Tested in multiple ways on both DB2 and Oracle with various ETL recovery job test scenarios:
Check for currently running job fails if ps command returns too long command string	Batch jobs	In Teradata environment the job parameters may exceed the length of the SAS string which is used to receive the output from a ps command in checking for the job just submitted already being executed. Result is that the same job may be executing concurrently cuasing one of them to abort with possibly partially processed data.	Recovery jobs were submitted twice within 5 seconds of each other and the observation was made that the second one aborted. This was confirmed in the rh/common/logs folder.