

Transaction Monitoring Systems How Back Office Processes Can Disable even the Best Ones

What started out as an apparently straight-forward Transaction Monitoring System Validation project took an interesting and cautionary turn at an international bank recently. The Project Team assembled for the task—as well as executive management at the Bank—expected that the Validation would discover some less-than-perfect data mapping from their core banking system to their Transaction Monitoring System. A completely new Compliance staff had reviewed the Bank’s unfamiliar (to them) Transaction Monitoring System, and could see that something wasn’t quite right. Wires were not appearing properly on reports, General Ledger account numbers were showing up instead of Customer Account Numbers, and there were unnecessary transaction codes, like Wire fees, clogging the system. It seemed like a simple, methodical task of documenting the current mapping and making appropriate changes. They determined to identify all the transaction codes and descriptions used by the Bank to eliminate redundant ones, as well as the codes that had no BSA/AML transaction monitoring value, like the fees.

However, half way through the project, some eye-opening facts hit the Team and sent them back to the drawing board. The re-mapping exercise turned up some remarkable and previously unknown facts that greatly affected the efficacy of their BSA/AML compliance program. Please note that this story is a true, but composite sketch of several projects at multiple banks.

Over 200 Transaction Codes

The Bank had over 200 transaction codes. This information was hard to obtain because the Back Office procedures were not well documented. It turned out that some of the transaction codes were native to the core banking system, and some were user-created. The Bank could not say which of the codes were used and which weren’t; whether they were used occasionally or frequently, and most importantly, what they meant. Even worse, the 200 transaction codes were *not* unique. Of the 200 transaction codes, over 20% of them had multiple transaction descriptions, some of them meaning totally different things. For instance an “CO” could mean *Cash from Vault*, *Teller Shortage*, *Split Deposit* or *Miscellaneous Cash Out*. They came from the same program in the core banking system, making it impossible to eliminate the *Teller Shortage* while keeping the *Split Deposit*.

Over 50 Types of Miscellaneous Credits

Among the 200 transaction codes were over 50 types of *Miscellaneous Credits*. They differed only by the Transaction Description field and contained descriptions such as *Reverse Overdraft*, *Stop Payment Charge*, *Partial Security Purchase*, and *Cash Check not On Us*. There were an almost equal number of *Miscellaneous Debits*, with the same problem of multiple transaction descriptions. In the test month, the Team found that there were close to 14,000 *Miscellaneous Credits* transactions. It was startling that with so many “real” transaction codes, that the Bank used the *Miscellaneous Credits (MC)* code so frequently.

In the Transaction Monitoring System reports, meanwhile, all the Compliance staff could see was the code *MC*. Some of the *Miscellaneous Credits* were for very large dollar amounts and clearly needed to be monitored for potentially suspicious activity, yet there was no way to determine the nature of the *MC* without researching the core banking system's files or working with Customer Support staff.

Lack of Data Entry Controls

When questioned about the over-abundance of transaction codes in general and the *Miscellaneous Credit* transactions in particular, the Back Office staff revealed that the core banking system enabled them to key in just about anything. Back Office staff could add a new transaction code, and any staff member could enter a description of his or her choice. There were no drop-down controls on the transaction codes, nor did the core banking system provide an automatic transaction description based on the entered transaction code. The Bank was using a very old version of the core banking system, and the implications for BSA/AML transaction monitoring were obvious.

Missing Wire Data

Digging even further, the Team discovered issues in the very important Wires department. As an international bank, the Bank did a large number of wires—generated via fax, the Internet and SWIFT. Because the original focus of the project had been on re-mapping, the Team had concentrated on getting the General Ledger Account Number out of the Customer Account Number field in the transaction monitoring system, along with other mis-mappings, such as the Beneficiary Name showing up in the Fourth Party field. The re-mapping was manageable. However, further testing revealed a disturbing fact. In testing Wires during the Validation, the faxes, Internet forms and SWIFT documents were compared to what was captured in the core banking system (This was to make sure data wasn't being truncated, and will be further explored in an upcoming article). The big surprise here was that in close to 50% of the wires, the Beneficiary Address wasn't entered into the core banking system. They were there on all the source documents, but for whatever reason, someone chose not to enter this important BSA/AML monitoring information into the core banking system. And as the Compliance Department knew, information not entered is information not monitored.

The Back Office of any organization is integral to the success of BSA/AML compliance. The Bank has one of the best Transaction Monitoring Systems on the market—well-known for its ability to monitor wires and flexible in its profiles and business logic units. However, no amount of re-mapping can make up for poor back office controls and an unsophisticated core banking system. Efforts are underway to introduce better controls, and the Bank has hired Certified Anti-Money Laundering Specialists (CAMS) under new and impressive Compliance leadership. But had this situation been left unchecked, the bank would be on its way to possible regulatory action and a Look-Back.