



To: City Manager John Baumgartner
From: Kristine Polian, Director of Finance
Subject: Staff Recommendation- Credit Card Processing Vendor
Date: May 7, 2019
cc: Assistant City Manager Michael Kramm; IT Director Ryan Smith

In March of this year, the City went live with its Utility Billing software, and several billing/ payment issues ensued. Included in those issues was the reoccurring credit card bill payment option failed to process. It was discovered the reoccurring credit card payment process was not functioning properly, and several customers were affected for at least two billing cycles. Staff began the process of identifying the issues specific to the reoccurring credit card payment problem and identified an issue with the vendor handling the credit card processing for payments.

Currently, the City uses BridgePay to handle the credit card processing (only two processing vendors can be used with the new accounting software, MUNIS). As such, when the reoccurring credit card payments were set up prior to going live with the new software, there was the indication from the vendor that nothing additional would need to occur in order to implement the reoccurring process; however, this was not the case. Each time the City would send a batch of utility payments to be paid via reoccurring credit card payment, BridgePay would reject the batch. This continued repeatedly until it was discovered by staff those payments were not actually being processed. After the issue was finally identified, City staff was notified that something additional did in fact need to occur in order to make the reoccurring payments process correctly.

Along with the current issue of reoccurring credit card payments, several other issues have been identified with the current processing vendor that has prompted staff to look at the other available processing vendor, OpenEdge. After an extensive comparative analysis on all factors of consideration, staff would like to recommend the City move to OpenEdge as the sole credit card processing vendor. Those reasons are as follows:

- OpenEdge is the sole processor/gateway involved in credit card processing, whereas BridgePay needs a processor, as they only serve as the payment gateway.

Administration is made easier with a vendor that handles the entire credit card processing function;

- OpenEdge can support credit cards with chips, where BridgePay does not, and has no definitive date when this function will be available. At this time, Visa and Mastercard will hold all merchants liable for any charges contested that could have been prevented by supporting the chip reader; the City has a moderate level of exposure currently and will continue to if the chip reading function is unavailable;
- The City will have to go through a process of getting the reoccurring credit card process functioning, regardless of which vendor is chosen; there is no time savings in staying with the current vendor;
- Anytime there is an issue with the current processing vendor, City staff cannot deal directly with the vendor, but must go through the new software company to deal with any issue. The recommended vendor allows direct access to resolve issues. This will result in a time savings for each issue, as well as less administrative burden on the City;
- Municipal Court will have to make a change from their current processing vendor, as it will no longer be supported by its software company, and the preferred processing vendor is OpenEdge. This will make for a more cohesive environment for IT, as there will be continuity with the processing vendor.
- OpenEdge will replace all existing credit card processing terminals at no charge;
- OpenEdge offers a decline-minimizer by updating customer credit card information every 24 hours (minimize declined payments due to expired cards);
- Staff has lost confidence in the current processing vendor after the software conversion issues surfaced;
- OpenEdge will save the City 3% annually on processing and interchange fees annually, which based on current volume, is approximately \$52K;