OF-3392 “Review days between Statement Generation and Past Due Notice”  
Testing

1. Must refer the reader to the Greg Wright-provided document:  
   Said document will provide needed background to help the reader to identify the customers needed for a testing run.
2. Identify a customer which has a nonzero balance,  
   AND  
   which has not ever received a statement before.  
   there should be some querying to do this, and the transcript-capture I did should provide a start on this, right?  
   In DEV-instance, identified customer 2002567196 as a customer who had no statement and who had a non-zero balance.
3. For any customer who has not ever received a statement,   
   we must choose a statement as-of date which is more than thirty days after the individual line-item due dates in the order,   
   and   
   demonstrate that the new code does not contribute the line-item’s amount towards the “overdue” amount printed on the customer statement.
4. Execute “LWX AR Consolidated Statement” Ebsuite program:  
   The extant log-and-output files, alongside the testing transcripts and notes, and these notes, will have to suffice for now.  
   Not sure whether or not this can be shown via screen shots? Shall have to be “fully online” to gather such together…
5. Capture the log and the output file (if there is one…)  
   There are log files for DEV environment, and for customer number 2002567196;  
   Also, the data have been saved into the “rstewar” schema’s “lwx-statement-data table-copies!”;  
   There are log files, output files, appropriately named; might want to paste the bodies of these into the Word document

OF-3393: “Address is Rejected in Statement Process”  
Testing

1. Identify a suitable customer which would have missing geocode data.  
   Use the following query to identify such customers:  
   select \* from (  
    SELECT loc.ADDRESS1,  
    loc.ADDRESS2,  
    loc.ADDRESS3,  
    loc.ADDRESS4,  
    loc.CITY,  
    substr(loc.STATE,1,2),  
    substr(loc.POSTAL\_CODE,1,12),  
   -- terr.TERRITORY\_SHORT\_NAME,  
    --  
    nvl(loc\_assign.location\_id,0) good\_geocode  
    --  
    , addr.cust\_account\_id  
    , party\_site.party\_id  
    , hca.account\_number customer\_number  
    FROM  
   -- AR\_LOOKUPS l\_cat,  
   -- FND\_TERRITORIES\_VL terr,  
   -- FND\_LANGUAGES\_VL lang,  
    HZ\_CUST\_ACCT\_SITES addr,  
    HZ\_PARTY\_SITES party\_site,  
    HZ\_CUST\_SITE\_USES csu,  
    HZ\_LOCATIONS loc,  
    HZ\_LOC\_ASSIGNMENTS loc\_assign  
    --  
    , hz\_cust\_accounts hca  
    WHERE  
   -- addr.CUSTOMER\_CATEGORY\_CODE = l\_cat.LOOKUP\_CODE(+)  
    1=1  
   -- AND l\_cat.LOOKUP\_TYPE(+) = 'ADDRESS\_CATEGORY'  
   --  
   -- AND loc.COUNTRY = terr.TERRITORY\_CODE(+)  
   -- AND loc.LANGUAGE = lang.LANGUAGE\_CODE(+)  
    AND addr.PARTY\_SITE\_ID = party\_site.PARTY\_SITE\_ID  
    AND csu.CUST\_ACCT\_SITE\_ID = addr.CUST\_ACCT\_SITE\_ID  
    AND csu.SITE\_USE\_CODE IN ('STMTS','BILL\_TO')  
    AND NVL(csu.PRIMARY\_FLAG, 'N') = 'Y'  
    AND loc.LOCATION\_ID = party\_site.LOCATION\_ID  
    AND loc.LOCATION\_ID = loc\_assign.LOCATION\_ID(+) -- n.b. this is an "optional join" OF-3393  
    -- EXTRA TESTING PREDICATES, NOT PRESENT IN THE ORIGINAL QUERY:  
    and loc\_assign.location\_id is null  
    and hca.cust\_account\_id = addr.cust\_account\_id  
    and hca.party\_id = party\_site.party\_id  
   --  
   -- The following two predicates are bound to input variables for the original program  
   -- cursor:  
   -- AND addr.cust\_account\_id = cn\_cust\_id  
   -- AND party\_site.PARTY\_ID = cn\_party\_id  
   ) where rownum < 31;
2. Execute a statement run for that customer by executing the “LWX AR Consolidated Statement” Ebsuite program.  
   I executed this program for the customer number 2000271783, in the DEV instance.
3. Capture the log and the output file (if an output file is indeed produced.)  
   There were two different runs, for the above customer, one **with the new code in place**, which produced a statement-data-file and log (which contained the indication that the geocode was missing, but which also indicated that statement-generation would proceed nonetheless.)  
   Corresponding log and data files’ names:  
   ebs-test-log-2000271783-20200309-new-code-missing-geocode-detected.txt  
   ebs-test-output-2000271783-20200309-new-code-missing-geocode-detected.txt  
     
   The other statement-run was performed **with the old code in place**: the only artifact produced was a log file, no statement generation took place, and the log data indicated that the program (again, the **old** version) was not going to produce a statement. Therefore, no statement-data file was produced either.  
   Corresponding log file name:  
   ebs-test-log-2000271783-20200309-old-code-missing-geocode-not-fixed.txt