**Code suggestions/enhancements (Bolt’s Desk)**

1. Please use “Optional Explicit” on all the modules that has not been specified. Check for the variables that might not have been declared before use. Declare the variables using the appropriate data types.
2. Please look for unused code in the project and remove them (for example I have identified functions CleanName, GetConfiguration, dhFirstDayInMonth, DeleteDir, SendEmailAuto, DeleteFile, NotOnTop, OnTop, text, GetTaskBarSize and GetLines (GetLines2 is used instead) Some of them are probably used for testing purposes and the calls to the functions commented in the code. It is better to remove them before deployment.
3. I am not able to find the AutoExec Macro in the access database file. Please confirm whether ap\_DisableShift and ap\_EnableShift gets executed at the client deployment machine on application startup. If not, suggest to remove this code/module.
4. Change the call to GetLines2 in LoopThroughFiles function so that it is called only once. Let the function return an Array and then assign the values to corresponding variables after the function call. This will open the text file only once. File handling is a costly operation that could slow down the application very much and it should be restricted as much as possible. Also close the file object as soon as the required information is read from the text file.
5. In all CloseinCRM functions, unassigned second parameter (“A”) is passed and is not used by the function. Please remove the parameter.
6. Please remove the hardcoded list of MCode values in GetCodeNumber function and place it as a constants in a separate module (for readability and maintenance)
7. Please move the hard coded values(TempBCLPosting and also throughout the application) for shared folders, file creation folders, URLs etc. based on the mode of deployment and frequency of change:
8. If the entire Access database will be replaced in the client deployment location irrespective of whether application code will be updated or not (only the change of URLs etc.), create a standard module containing the corresponding constants and update the values there. The constants declared in the modules will be accessed in the code and hardcoding of values in the application code will be eliminated..
9. If Access database will not be replaced and only the URLs etc need to be changed, create a ini file (values would be read from the file based on the field property name) or txt file(each line would contain the field order and would be used in the code) and just update the values in the file. No changes in the access database would be required. The code would simply read the data from the file accordingly and do the processing as before (But it is important that users don’t tamper the file, otherwise the application will not work as expected). Either way hardcoding of values in the code will be eliminated.
10. Use instr function to check for the existence of a string eliminate multiple If conditions using OR. For example If browser.Title = "\*Login" Or browser.Title = "Login" Then , If browser.Title = "\*Accounts Receivable" Or browser.Title = "Accounts Receivable" Then etc.
11. Duplicate code in main function and in the Goto Condition BrowserErrorReRun - code can be put in a common function and just the function can be called at two places. BrowserErrorReRun condition was noticed for only two circles: Haryana and Punjab. Please consider whether this could be considered for the remaining circles as I do not consider much of an overhead.
12. Error handling and logging - Reconciliation process. Suggest to log all the processed transactions with status in a separate log folder in text file in a suitable format with new log file created every day. This would be useful instead of manual reconciliation to determine unprocessed records because of any error or application termination. (***Priority to be considered***)
13. Instead of empty If condition - If Forms![Autobots]![dupcheck].Value = "Dup" Then, it can be combined as If Forms![Autobots]![ReasonError].Value = "" and Forms![Autobots]![dupcheck].Value <> "Dup" Then
14. The following minor differences has been observed in BCLPostings function for all the circles:

**Delhi**: control name:beID,value=1130489

*Call browser.FindElement("a", "uiname=MUM").Click (should it be DEL)? for few other circles as well it is MUM*

**Gujarat**: control: ("select", "name=env").Select("vel1\_PE"), some code related to Accounts Receivable

**Haryana:** BrowserErrorRerun code

**MnG:** no differences observed

**MP:** no differences observed

**Mumbai**: not much except one variable name reversed errorreason

**Punjab:** control: name=beID,value=1130449, some code related to Accounts Receivable, BrowserErrorRerun code

**Rajasthan:** control: ("select", "name=env").Select("vel1\_PE"), control: name=beID,value=1130459").

**UPE**: control: name=beID,value=1130609"

control: ("textarea", "name=memo").InputText(Adreason & "\_" & SrNum & "\_" & "CBO")

**UPW**: control: name=beID,value=1130619

By using appropriate constants, we would be able to create single generic function called BCLPostings to handle functionality of all the circles. (***Request to handle this based on priority***)

1. Only difference noticed in all CloseinCRM functions is hardcoding of 4 URLs. Please declare these URLs as constants so that only one function is required instead of as many functions as the number of circles. We can have a small GetURLs function that would return the corresponding URLs on passing the Circle code. BCLPostings and CloseinCRM for all circles will be available in just a single module.
2. We can eliminate the calls to these 2 functions in a Big Select case for all circles in LoopThroughFiles function by simply passing the circle code as a parameter. The lines of code would be reduced very much because of the above 3 steps and we can expect improved performance.
3. Please indent the code.