

API Module

Module specification format:

|  |
| --- |
| XYZ  *Overview:* explain the rationale and responsibility of XYZ |
| *Public interface*: here you will list interface operations documented as follows: |
| *Operation header*: returned-value operation-name (list of parameters with names and types)  *\*Parameters (optional):* unless it is not clear from the header, describe parameters  *Description:*  Describe the effects of the operation (what the operation does) in terms of parameters, returned value and whatever else you need to explain in order to explain the meaning of the operation; Normal behaviour of the operations is explained first before the abnormal behaviours are explained.(handled by assertions and exceptions) |

|  |
| --- |
| **Read Individual Field**  Module description:  The module contains function which is needed 'to be read the components of a field from the BOTLAIDS file. The components that are specially taken care of are name required field, preloaded field data, field max size and field title. |
| Public Module: |
| **VOID Parse\_Field (fileStream As textstream);**  Description:  This function will read a whole field, and store all the relevant information into a field object. The function reads the field sequentially top-down. |
| **INTEGER Extract\_Number (rawLine As String)**  Description:  The function will extract the number in the string between the left and right delimiter. |
| **VOID Read\_Category\_Message (categoryCounter As Integer, fileStream As textstream, mapObject() As categoryMapping)**  Description:  This function will read the content inside a message category. The content are store inside an array of categoryMapping |

|  |
| --- |
| **Read Individual Segment**  Module description:  The module contains functions which are needed to be read the components of a segment from the BOTLAIDS file. The components that are specially taken care of are namely segment name, required segment, and the fields inside a segment. |
| Public Module: |
| **VOID Parse\_Segment (fileStream As textstream)**  Description:  The function will read a whole segment from the BOTLAID text stream, and store all the relevant component of a segment into a segment object. The function read the segment in sequential top-down manner. |
| **VOID Parse\_Segment\_Header (numOfField As Integer, outputArray() As String, complusory As Boolean, name As String, fileStream As textstream)**  Description:  The function reads the header section of a segment and returns the relevant information after the function ends. |

|  |
| --- |
| **Read Individual BOTLAID**  Module description:  The module contain functions which are needed to be read the components of a BOTLAID report file generate from the BOTLAID function in the batch manager. |
| Public Module |
| **VOID Parse\_BOTLAID (fileStream As textstream)**  Description:  The function will read the whole BOTLAID report file in a orderly top-down manner. The report generation information, classic transaction template and where write information of the BOTLAID report file are not captured in this function. |
| **CampaignObject Parse\_BOTL\_Header (fileStream As textstream)**  Description:  The function will read the header portion of the BOTLAID report, and store the relevant information into a campaign object. |
| **Parse\_Single\_message (rawLine As String, delimiter As String, code As String, description As String, Optional frontString As Variant)**  Description:  The function will read in the string which contains two information to be stored, separated by a delimiter. The function also contains an optional parameter (front string) to get rid of the unnecessary front string in the front. |
| **BOOLEAN Set\_Toggle\_By\_Statement (rawLine As String, stringPositive As String, Optional stringNegative As Variant)**  Description:  The function will read the string and return true/ false depending on the positive and negative template statements. |
| **AgeRange Find\_Age\_Range (rawLine As String)**  Description:  The function will read a string and return the minimum and maximum age range extracted in the string. |
| **Product() Parse\_Product\_Information (fileStream As textstream)**  Description:  The function will read a whole product in the header portion of BOTLAID report. All the relevant information will be stored as a product and return at the end of the function. |

|  |
| --- |
| **Generate Templates**  Module description:  The module contains functions which are required to generate the template for a particular BOTLAID report selected from the user. |
| Public Module: |
| **Generate\_Template ()**  Description:  This function will be the main function used to generate the templates based on the BOTLAID report which user selects. If the campaign allows for dependents, the user will also be prompted to enter the maximum number of dependents allow for that templates, and use the input to generate that number of dependents. |
| **Integer Set\_Maximum\_Dependent ()**  Description:  This function will pop up the dialogue box to ask user for the maximum number of dependents. |
| **Write\_Template\_Data (campaign As CampaignObject, maxDependent As Integer)**  Description:  This function is the main function which does the main job of organising the segments to be written on the spread sheet. The function will also carry out data validation for those which only takes in certain code. The sequence of generating the templates is to generate the campaign generic information first, then generate each person's templates in the order of policy payer, policy holder, main insured, spouse and lastly dependents. |
| **Write\_Person\_Details (segmentList() As String, campaign As CampaignObject, ParamArray additionalSegment() As Variant)**  Description:  This function aims to generate all the segment and field for a typical person's details required. There is also a paramArray to take in the additional segment that is needed on top of the typical person. For example, a policy payer will need to display fields for payment information. |
| **Write\_Segment\_Header (segment As SegmentObject)**  Description:  This function will write the segment header and the fields inside the segment. |
| **Write\_Field\_Headers (segment As SegmentObject)**  Description:  This function will write all field names within a segment onto the templates excel sheet. The fields will not write fields with fixed values and in the ignored list. The fields will also be formatted. |
| **Boolean Is\_In\_Array (item As Variant, referenceArray As Variant)**  Description:  This function check if a item is inside the array. Both parameters are of variant data type to cater to the various type of parameter that could be pass in. |
| **Boolean Has\_Elements (pArray() As categoryMapping)**  Description:  This function will check if there is any element inside the array. |
| **Generate\_Validation ()**  Description:  This function will write all the validation list options on the side of the spread sheet and put the data validation restriction onto the respective columns. If the validation list options have been written before, it will not be written again. The validation list options will be hidden so that the users who will be using the templates will not need to see the unnecessary information. |
| **Build\_Validation\_List (referenceColumn As Integer, Optional writeColumn As Integer, Optional mappingSize As Variant)**  Description:  This function will build put the column data validation list restriction. |
| **Write\_Address\_Reference\_Segment ()**  Description:  This function writes the address reference segment and its field headers onto the templates. |
| **Generate\_Address\_Reference\_Validation ()**  Description:  This function will generate the data validation for the segment "Address Reference". It uses all the person type in the templates as the data validation list options. |

|  |
| --- |
| **Generate BOTL intermediate file**  Module description:  The module contains functions which are need to read the whole excel spreadsheet filled with enrolment details. The modules also contain functions that will write the entry to a user-defined file name and destination path. |
| **Generate\_BOTL\_intermediate ()**  Description:  The function is the main function used to generate the whole BOTL intermediate file. |
| **Integer Last\_Column\_Of\_One\_Row (referenceRow As Integer)**  Description:  The function will return the last column of an input row number. |
| **Integer Last\_Row\_Of\_One\_Column (referenceColumn As Integer)**  Description:  The function will return the last row of an input column number. |

|  |
| --- |
| **Read Individual Record**  Module description:  The module contains functions which are needed to read the fields inside a raw entry in the completed form. |
| Public Module: |
| **Parse\_Record (referenceRow As Integer, campaign As CampaignObject, writeStream As textstream)**  Description:  This function reads a row of record and writes the information in the record into the text file. The sequence number need in the BOTL entry is derived from the row number that the entry is keyed in. |
| **Boolean Segment\_Not\_Empty (numberOfField As Integer, referenceRow As Integer)**  Description:  This function test if the any of the field within the segment has been filled. Return true if there is at least 1 field filled. Return false if none is filled. |
| **Set\_Field\_Content\_To\_Record (numberOfField As Integer, referenceRow As Integer)**  Description:  The function will read the value filled by the user on the spread sheet and set the value in the respective segment in the record object. |
| **Set\_Fixed\_Value (campaign As CampaignObject)**  Description:  This function will pre-set some of the values determined by the users. |
| **Set\_Value\_Into\_Field (campaign As CampaignObject, trackingSegment As String, trackingField As String, setValue As Variant)**  Description:  The function is a intermediate function for Set\_Fixed\_Value. It will set the value of one individual field. |
| **Boolean Is\_Inside\_Category\_Mapping (code As String, mapping() As categoryMapping)**  Description:  This function will check if a code is inside the category mapping array. |
| **Write\_Segment\_To\_File (writeStream As textstream, Optional segmentName As Variant)**  Description:  The function will write the segment with all the fixed value, pre-set value and already filled value into the output file. The function will also check if there is any missing value in the compulsory field in the segment. The function will write the segment out in a line. |
| **Integer Find\_Number\_Of\_Field\_In\_Segment (column As Integer)**  Description:  This function will return the number of fields in a segment. |
| **Initialise\_Address\_Reference\_Array (lastColumn As Integer)**  Description:  The function initialises the address reference array in the array by setting the size of the array. It reads the Address Reference options available from the templates. The options will always be next to the last row of the (unhidden) templates. Then we will be able to track the maximum number of address reference array size. |
| **Parse\_Address\_Reference (referenceRow As Integer)**  Description:  This function parses the address reference segment. The information parsed will be set inside the address reference array and it is updated in the record also. If user enter something into address reference, then we will check the name entered against the address reference array, and copy the address ID and address Type Code into the entry of that person. If the user enter the address and addrtype, then we will capture the addrtype information at that point of entry and put it inside the address reference array. |
| **Parse\_Address\_Type\_Segment (personType As String, inputAddressTypeCode As String)**  Description:  The function will read the address type segment from the spread sheet and update the address reference array. For policy payer and policy holder, we will update the policy payer address type in the payment segment and policy holder in the master segment. |