Test Log

Field

Test Case: Field (1)

Date designed: 8 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains one compulsory field, which is specific.

Input:

*[4] - Transaction Code*

*Field is Required*

*Must be "NEW"*

*Data type is string[3]*

*-----*

\*Data is stored inside “Field (1).txt” inside *Test Files* folder.

Expected behavior: Code will run without error. Output is “NEW”.

Test Case: Field (2)

Date designed: 8 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains one compulsory field, which is not specific. Input from active worksheet.

Input:

*[3] - Last Name (Surname)*

*Field is Required*

*Data type is string[50]*

*-----*

\*Data is stored inside “Field (2).txt” inside Test Files folder.

Expected behavior: Code will run without error. A new variable is declared.

Test Case: Field (3)

Date designed: 11 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains one non-compulsory field, which is specific from a list. List is given in the Field segment.

*[8] - Title*

*Data type is int*

*Value in Message Category 3 (Caller Title)*

*100995 - Dr*

*100728 - Mdm*

*100519 - Miss*

*100855 - Mr*

*100858 - Mrs*

*100516 - Ms*

*100568 - Prof*

*-----*

\*Data is stored inside “Field (3).txt” inside Test Files folder.

Expected behavior: Code will run without error. A new mapping array will be formed. A new variable is declared.

Test Case: Field (4)

Date designed: 11 June 2012

Test Objective: A valid entry of typical field inside a segment. Input is an account ID field which can be represented by macro.

Input:

*[2] - Account ID*

*Field is Required*

*Data type is string[9]*

*Primary Account ID. May use macro representing account id.*

*-----*

\*Data is stored inside “Field (4).txt” inside Test Files folder.

Expected behavior: Code will run without error. The variable that is used to store the account ID will refer to global variable, e.g. $\_acct1.

Segment

Test Case: Segment (1)

Date designed: 13 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains a compulsory segment, and one compulsory field.

Input:

*BEGIN|[2]|[3]|NEW|SG|[6]|[7]*

*SegmentKey: BEGIN*

*Segment is required*

*--------------------*

*[2] - RunID*

*Field is Required*

*Data type is string[20]*

*Identifies the BOTL run*

*--------------------*

Expected behavior: Code will run without error. Output is “BEGIN|[2]|[3]|NEW|SG|[6]|[7]”.

Test Case: Segment (2)

Date designed: 13 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains a non-compulsory segment, with 2 compulsory fields inside. One of the fields contains a category.

Input:

*SUPPRESS|[2]|[3]*

*SegmentKey: SUPPRESS*

*--------------------*

*[2] - Account ID*

*Field is Required*

*Data type is string[9]*

*May use macro representing account id.*

*-----*

*[3] - Reason*

*Field is Required*

*Data type is int*

*Value in Message Category 37 (Suppression Code)*

*100892 - Fulfillment via Email*

*100935 - No Call - Home*

*100725 - No Call - Home and Office*

*100517 - No Call - Office*

*100642 - No Promotional Mail*

*--------------------*

Expected behavior: Code will run without error. Output is “*SUPPRESS|[2]|[3]*”.

Test Case: Segment (3)

Date designed: 13 June 2012

Test Objective: A valid entry of typical field inside a segment. Input contains a compulsory segment, with a mixture of compulsory and non-compulsory fields inside. some of the fields contains a category.

Input:

*PRODUCT|AMPA09|[3]|[4]|[5]|[6]|[7]*

*SegmentKey: PRODUCT-AMPA09*

*Segment is required*

*--------------------*

*[2] - Product ID*

*Field is Required*

*Must be "AMPA09"*

*Data type is string[6]*

*Protection Growth Plan*

*-----*

*[3] - Coverage Code*

*Field is Required*

*Data type is int*

*Who is covered - Message category COVERAGE*

*1686 - Main Insured and Dependents*

*1682 - Main Insured and Family*

*1681 - Main Insured and Spouse*

*1680 - Main Insured Only*

*1684 - Spouse and Dependent*

*1683 - Spouse Only*

*-----*

*[4] - Number of Eligible Dependents*

*Data type is int*

*-----*

*[5] - Premium Override Amount*

*Data type is double*

*-----*

*[6] - Premium Override Expire*

*Data type is string[5]*

*If present, format MM/YY.*

*-----*

*[7] - Premium Override Reason*

*Data type is int*

*Value in Message Category SW (S6 Premium Override Reasons)*

*100838 - Goodhealth Premium*

*100612 - Incorrect Date Of Birth*

*100619 - Incorrect Prem (at Conversion)*

*100773 - Incorrect Prem (PM vs Mkt Mat)*

*100766 - Incorrect Rating Factor*

*100484 - Incorrect Sex Code Used*

*0 - None*

*100491 - Staff Premium*

*--------------------*

Expected behavior: Code will run without error. Output is “*PRODUCT|AMPA09|[3]|[4]|[5]|[6]|[7]*”.

BOTLAIDS

Test Case: SG12000601

Date designed: 18 June 2012

Test Objective: A typical working file.

Input: [BOTLAID.NEW.SG12000601.log](../System%206%20Renovation%20System%20Files/BOTLAID.NEW.SG12000601.log)

Expected Behavior: Code will run without error.

Test Case: SG12019401

Date designed: 18 June 2012

Test Objective: A typical working file.

Input: [BOTLAID.NEW.SG12019401.log](../System%206%20Renovation%20System%20Files/BOTLAID.NEW.SG12019401.log)

Expected Behavior: Code will run without error.