# 

**PRODUCT - Country Model Bank Development & Client Specific Development**

**TECHNICAL SPECIFICATION DOCUMENT**

**HUSBANKEN**

**CR0015T**

Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author / Changed By | Status | Change Description |
| 0.1 | 03.10.2022 | Dhivyambika T | Initial | Drafted |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Sign Off History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Temenos Authorisation | Client Authorisation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Management Summary 4](#_Toc41498781)

[2. Functional Overview 4](#_Toc41498782)

[3. Operational Overview 5](#_Toc41498783)

[4. Reusable Components 5](#_Toc41498784)

[5. Design Analysis and Solution 6](#_Toc41498785)

[6. New / Amended Files 6](#_Toc41498786)

[7. Technical Overview 9](#_Toc41498787)

[8. New Tables 9](#_Toc41498788)

[9. Subroutines 10](#_Toc41498789)

[10. Appendix 11](#_Toc41498790)

[11. Accounting 11](#_Toc41498791)

[12. Limits 11](#_Toc41498792)

[13. COB Processing 11](#_Toc41498793)

[14. Assumptions 11](#_Toc41498794)

[15. Dependencies/Exclusions 11](#_Toc41498795)

[16. Testing Notes 12](#_Toc41498796)

[17. Packaging / Installation 12](#_Toc41498797)

[18. Data Migration 12](#_Toc41498798)

# Management Summary

## Background

|  |  |  |
| --- | --- | --- |
| **Gap Reference** | **Description** | **Summary** |
| BRD01-01 | Attest Disbursement | New local application to Attest the Disbursement |
| BRD01-02 | Approve Attested Disbursement | New Version to Approve/Authorise the Attested Disbursement |
| BRD01-03 | Amend PO Disbursement | Amend PO to update the KID reference and or add additional payment information |
| BRD01-04 | Letter of disbursement | Generate XML for Husbanken middleware to produce Letter of Disbursement |

## Modification Overview

This document details about the technical solution of Attest disbursement and Disbursement letter on the Payment initiation done for disbursement

## Reference Documents

|  |  |  |
| --- | --- | --- |
| Reference Document | Description/Path | Author, Company |
| FSD |  | John Paul |
|  |  |  |

### Reference Number of the Requirement (ODR) which is dependent on this Requirement

## Glossary (Optional)

This section provides glossary & descriptions that are used in this document.

|  |  |
| --- | --- |
| Glossary | Description |
| HB | Husbanken |
| LT | Local Table |
| LRT | Local Reference Field |
| BRD | Business Requirement Document |
| FSD | Functional Specification Document |
| IRD | Interface Requirement Document |
| PO | Payment Order |
| TPH | Temenos Payment Hub |
| COS | Composite Screen |
| TAB | Tabbed Screen |
| HM | Helptext Menu |

# Functional Overview

## Existing Functionality

The existing HB Loan Disbursement has been done manually through versions with TPH related functionalities solution along with additional L3 developments.

## New Functionality

Proposal of New functionality has been given:

The Payment for HB Loan Disbursement should be done only after the Attest Disbursement process is completed.

**Attest Disbursement**

* Create a new record in an Attest Disbursement template with @ID as payment order id.
* An enquiry to be created to display the unauthorised payment order record with view and edit icon in the enquiry.
* Attest Disbursement record should be authorised after document check is completed and validation needs to be done to check if all the documents check is completed.
* After authorising the attest disbursement record, local field in payment order needs to update as “Disbursement Attested”

**Authorise Attested Disbursement**

* An enquiry should be created by copying the core enquiry PO.UNAUTH.TXN to list unauthorised payments with “Disbursement Attested” status.
* PO’s can only be disbursed if there is sufficient Budget availability
* PO’s can only be authorized if the user is other than the one that attested the PO.

**Loan Manager Amend Payment Order**

* Create a new version to amend the Payment Order.
* Make all the field as no imputable fields other than HUS.KID.NO and Payment Information (Free Text field).
* Validation needs to be done against the HUS.KID.NO based MOD 10 and MOD11 algorithm.

**Letter of Disbursement**

* The Authorisation of the Attested Disbursement will result in the creation of the necessary XML data being sent to the Husbanken Document Service

# Operational Overview

NA

# Reusable Components

## Pack Reusability

NA

### Reusable – (Yes/No)

### Reason

NA

## Reusable Components Table

NA

# Design Analysis and Solution

NA

# New / Amended Files

NA

## Local Table

### Local Table Name

|  |  |  |
| --- | --- | --- |
| No. | Field | Definition |
| 1 | Number | HUS.ATST.STATUS |
| 2 | Description | Attest Status |
| 3 | Short Name | HUS.ATST.STATUS |
| 4 | Max. Char | 35 |
| 5 | Min. Char |  |
| 6 | Char Type | “” |
| 7 | Vetting Table |  |
| 8 | Application | NA |
| 9 | Enri. Field | NA |

## LOCAL.REF.TABLE

|  |  |  |
| --- | --- | --- |
| Application | Local Ref | Association |
| PAYMENT.ORDER | HUS.ATST.STATUS |  |

## Enquiry

### HUS.ATTEST.PAY.ORDER

Create a new enquiry to list the unauthorised payment order. Include the selection fields as in the core enquiry PO.UNAUTH.TXN.

|  |  |
| --- | --- |
| FIELD.NAME | VALUE |
| PAGE.SIZE | 4,40 |
| FILE.NAME | PAYMENT.ORDER$NAU |
| FIXED.SELECTION |  |
| SELECTION.FLDS.1 |  |
| SEL.LABEL.1 |  |
| REQUIRED.SEL.1 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Label in Output** | **Field Name** | **Operation** | **Column** | **Conversion** | **Other Attributes** |
| Payment Order ID | PO.ID | @ID | 1 |  |  |
| Case No | HUS.CASE.NO | HUS.CASE.NO | 2 |  |  |
| Customer ID | ORDERING.CUSTOMER | ORDERING.CUSTOMER | 3 |  |  |
| Customer Name | ORD.CUST.NAME | ORD.CUST.NAME | 4 |  |  |
| Customer National ID | ORDERING.CUSTOMER |  | 5 | L CUSTOMER,HUS.REGISTER.ID |  |
| Amount | PAYMENT.AMOUNT | PAYMENT.AMOUNT | 6 |  |  |
| Loan Product | PAYMENT.ORDER.PRODUCT | PAYMENT.ORDER.PRODUCT | 7 |  |  |
| Beneficiary Name | BENEFICIARY.NAME | BENEFICIARY.NAME | 8 |  |  |
| Beneficiary Account | BENEFICIARY.ACCOUNT.NO | BENEFICIARY.ACCOUNT.NO | 9 |  |  |
| Execution Date | PAYMENT.EXECUTION.DATE | PAYMENT.EXECUTION.DATE | 10 |  |  |
| Value Date | CREDIT.VALUE.DATE | CREDIT.VALUE.DATE | 11 |  |  |
| Initiation Type | ORDER.INITIATION.TYPE | ORDER.INITIATION.TYPE | 12 |  |  |

**Drilldown and other special attributes:**

|  |  |
| --- | --- |
| Enquiry Name | VER.NAME(PAYMENT.ORDER) S PaymentOrderID |
| Label Field | @ID |
| Nxt Desc | View[view.gif] |
| Enquiry Name | VER.NAME(EB.HUS.ATST.DISBURSE,INPUT) I PaymentOrderID |
| Label Field | @ID |
| Enquiry Name | Edit[edit.gif] |

### HUS.AUTH.ATTEST.PAY.ORDER

Create a new enquiry copying the core enquiry PO.UNAUTH.TXN and with the Fixed selection as “HUS.ATST.STATUS” EQ “Disbursement Attested”.

|  |  |
| --- | --- |
| FIELD.NAME | VALUE |
| PAGE.SIZE | 4,40 |
| FILE.NAME | PAYMENT.ORDER |
| FIXED.SELECTION | CURRENT.STATE EQ ‘AwaitingExtSubmit’ AND HUS.PAY.STATUS EQ ‘Disbursement Attested’ |
| SELECTION.FLDS.1 |  |
| SEL.LABEL.1 |  |
| REQUIRED.SEL.1 |  |

### HUS.PO. AMEND

Create a new enquiry copying the core enquiry PO. AMEND and change the amend version as PAYMENT.ORDER,HUS.AMEND

|  |  |
| --- | --- |
| FIELD.NAME | VALUE |
| PAGE.SIZE | 4,40 |
| FILE.NAME | PAYMENT.ORDER |
| FIXED.SELECTION |  |
| SELECTION.FLDS.1 |  |
| SEL.LABEL.1 |  |
| REQUIRED.SEL.1 |  |

**Drilldown and other special attributes:**

|  |  |
| --- | --- |
| Enquiry Name | VER.NAME(PAYMENT.ORDER,HUS.AMEND) I PaymentOrderID |
| Label Field | @ID |
| Enquiry Name | Edit[edit.gif] |

## Context Enquiry

### [Context Enquiry Name]

NA

## Version

### EB.HUS.ATST. DISBURSE,INPUT

Create a new version with L3 table HUS.ATTEST.PAY.ORDER with all the fields. Below is the sample Layout.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

NO.OF.AUTH>1

### EB.HUS.ATST. DISBURSE,APPROVE

Create a new version with L3 table HUS.ATTEST.PAY.ORDER with the below fields and

NO.OF.AUTH as 0

Graphical user interface, application

Description automatically generated

### PAYMENT.ORDER,HUS.AMEND

Create a new version by copying the existing version PAYMENT.ORDER,HUS.INTIATION.CREATE and make all the field as no input table fields and make the below fields as imputable fields.

NO.OF.AUTH>1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Description** | **Actual T24 field name** | **Mandatory** | **No input field** | **Remarks** |
| Kid Number | HUS.KID.NO |  |  |  |
| REMITTANCE.INFORMATION | Payment Information |  |  | Free Text field |

## Composite Screen

### HUS.PO. AMEND

Create a composite screen by copying the PO. AMEND composite screen and authorise the record.

## Tabbed Screen

### NA

## HelpText Menu

### HUS.LOAN.MGR.MENU

|  |  |
| --- | --- |
| Field Name | Field Value |
| VERSION | EB.HUS.ATST.DISBURSE,APPROVE |
| Description | Authorise/Delete Attested Disbursement (PO) |

### HUS.PO.AMEND

|  |  |
| --- | --- |
| Field Name | Field Value |
| Application | COS HUS.PO.AMEND |
| Description | Amend Payment Order (Husbanken) |

## I-DESCRIPTORS

[A

## STANDARD.SELECTION

NA

## Data Records

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Table Name | Record id | Detail |
| 1 | EB.ERROR | EB-HUS.DISB.ATST.USER | Disbursed User cannot authorise the Payment |

# ORorease y previous Dependant Packs or Developments related to this devlopment

# Technical Overview

NA

## Architecture

## Communications (Optional)

NA

# New Tables

## HUS.ATST.DISBURSE

### Table Properties

|  |  |  |
| --- | --- | --- |
| No. | Property | Definition |
| 1 | Name | HUS.ATST.DISBURSE |
| 2 | Product | EB |
| 2 | File Type | H |
| 3 | Classification | INT |

### Table Layout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **DESCRIPTION** | **MAX.CHAR** | **CHAR.TYPE** | **MV** | **Other /Application vetting/Virtual Table** |
| @ID | @ID | 6 | Alphanumeric |  |  |
| PAYMENT.ID | Payment Reference |  |  |  |  |
| CASE.ID | Case Id | 25 | Alphanumeric |  |  |
| ARRANGEMENT.ID | Arrangement Id | 25 | Alphanumeric |  |  |
| BORR.NAME | Borrower (Name) | 25 | Alphanumeric |  |  |
| BORR.NATION.ID | Borrower National Id | 25 | Alphanumeric |  |  |
| PROD.DESC | Product Description | 25 | Alphanumeric |  |  |
| CADASTRAL.ID | Cadastral ID | 25 | Alphanumeric | XX. |  |
| BLD.STATUS | Building Status | 25 | Alphanumeric |  |  |
| COLL.TYPE | Collateral Type | 25 | Alphanumeric |  |  |
| COLL.CODE | Collateral Code | 25 | Alphanumeric |  |  |
| GURANTOR.NAME | Guarantor Name | 25 | Alphanumeric |  |  |
| GUARANT.NATION.ID | Guarantor National ID | 25 | Alphanumeric |  |  |
| GUARANT.AMT | Guarantee Amount | 25 | Alphanumeric |  |  |
| COLL.LINK.ID | Collateral Link ID | 25 | Alphanumeric |  |  |
| ALLOC.AMT | Allocation Amount | 25 | Amount |  |  |
| LEGAL.RIGHT.REF | Legal Right Reference | 25 | Alphanumeric |  |  |
| DISBURSE.AMT | Disburse Amount | 25 | Amount |  |  |
| FULL.OR.PART | Full/Partial |  |  |  |  |
| BENE.NAME | Beneficiary Name | 25 | Alphanumeric |  |  |
| BENE.ID | Beneficiary | 25 | Alphanumeric |  |  |
| ACC.NO | Account Number | 25 | Alphanumeric |  |  |
| KID.NO | KID Number | 25 | Alphanumeric |  |  |
| PAY.INFO | Payment Information | 50 | Alphanumeric |  |  |
| INTERNAL.FLD | Internal Notes Only | 100 | Text |  |  |
| ATTEST.BY | Attested By | 30 | Alphanumeric |  |  |
| DATE | Date | 30 | Alphanumeric |  |  |
|  |  |  |  |  |  |
| REG.MORT.DOC | Registered Mortgage Doc |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| DEC.GUARNT.MORT.DOC | Declaration Guarantee Mort Doc |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| LOAN.AMT.DEC.DOC | Loan Amount Declaration doc |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| REGISTERED.RIGHT | Registered Right |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| AGREEMENT.LEASEHOLD | Agreement Leasehold |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| INSURANCE.PLACE | Insurance in Place |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| RIGHT.OF.ENTY.REG | Right of Entry Registered |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| AGREE.WITH. MUNICIPAL | Agreement with municipal |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| DEC.OF.RIGHTS | Declaration of rights |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| LOAN.AGREEMENT | Loan Agreement |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| GRANT.AGREEMENT | Grant Agreement |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| CONSENT.ON. TERMS | Consent on Terms |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| INVESTMENT.BUDGET | Investment Budget |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| MUN.COUNCIL.DEC | Municipal Council Decision |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| STATE.ADMIN | State Administrator |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| SELF.REPORT | Self-Report |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
|  | Paid Equity |  |  |  | Documents Pending Verification  Document Check – Not Required  Documents Checked and Attested |
| PAID.EQUITY |  |  |  |  |  |
|  |  |  |  |  |  |

### Field Checks

|  |  |  |
| --- | --- | --- |
| S.No | Name of the field | Check’s to be done |
|  |  |  |
|  |  |  |

### Cross Validation Checks

NA

### Other Processing

**Check function:**

NA

1. **BEFORE.AUTH.WRITE:**

NA

1. **AFTER.AUTH.WRITE:**

NA

### INSERT Files

NA

# Subroutines

## HusValPayUser

|  |  |
| --- | --- |
| Property | Specification |
| Type | S |
| Attached To | PAYMENT.ORDER,NEW |
| Attached As | Validation Routine |
| Dependency | NA |
| Description | Routine to validate the user authorising the Payment order should be different from the user who authorise the attested Disbursement record. |
| Arguments – IN |  |
| Arguments – OUT |  |
| Prelim Conditions | NA |
| Subroutine Flow | 1.Read the current user and store in a variable.  2.With the Payment order ID read the EB.HUS.ATTEST.DISBURSE record and read the user and store it in a variable.  3.If the user are same system should raise the error “Disbursement Attested user cannot authorise the Payment”.  4.End of the program. |
| Special Instructions | NA |

## HusUpdPayInfo

|  |  |
| --- | --- |
| Property | Specification |
| Type | S |
| Attached To | PAYMENT.ORDER,HUS.AMEND |
| Attached As | Auth routine |
| Dependency | NA |
| Description | Routine to update the Payment Information value in attest disburse template |
| Arguments – IN |  |
| Arguments – OUT |  |
| Prelim Conditions | NA |
| Subroutine Flow | 1.Read the value from Payment Information field in current Payment order record and store in a variable.  2.With the Payment order ID read the EB.HUS.ATTEST.DISBURSE record.  3.Update the value in the field PAY.INFO in EB.HUS.ATTEST.DISBURSE record.  4.End of the program. |
| Special Instructions | NA |

## HusAuthAtstDisburse

|  |  |
| --- | --- |
| Property | Specification |
| Type | S |
| Attached To | PAYMENT.ORDER,HUS.INITIATION.CREATE |
| Attached As | Auth Routine |
| Dependency | NA |
| Description | This routine is to update EB.HUS.ATST.DISBURSE record with @ID as current payment order ID in INAU status |
| Arguments – IN | NA |
| Arguments – OUT | NA |
| Prelim Conditions | NA |
| Subroutine Flow | 1. Get the currentRecord id.      |  |  | | --- | --- | | **EB.HUS.LA.CASE** | **EB.HUS.ATST.DISBURSE** | | Current record | PAYMENT.ID | | @ID | CASE.ID | | AA.ARRANGEMENT>@ID | ARRANGEMENT.ID | | PARTY.NAME | BORR.NAME | | REGISTER.ID | BORR.NATION.ID | | AA.ARRANGEMENT> PRODUCT:1 AA.PRODUCT> DESCRIPTION | PROD.DESC | | CADASTRE.ID | CADASTRAL.ID | | COLLATERAL>BLD.STATUS | BLD.STATUS | | COLLATERAL>COLLATERAL.TYPE | COLL.TYPE | | COLLATERAL>COLLATERAL.CODE | COLL.CODE | | COLLATERAL.RIGHT>COLLATERAL.RIGHT@ID>REPL.COMP.ID  CUSTOMER>NAME.1 | GURANTOR.NAME | | COLLATERAL.RIGHT>COLLATERAL.RIGHT@ID>REPL.COMP.ID | GUARANTOR.NATION.ID | | COLLATERAL.RIGHT>REPL.AMT | GUARANTEE.AMT | | COLLATERAL.RIGHT.ID | COLL.LINK.ID | | ALLOCATION.AMT.1 | ALLOC.AMT | | EB.HUS.LA.CASE>COLLATERAL.RIGHT@ID  COLLATERAL.RIGHT>COLLATERAL.RIGHT@ID>  SEC.DOC.YEAR(Ex 2010)  SEC.DOC.NR(Ex 12345)  SEC.DOC.OFF.NR(Ex 201)  LEGAL.RIGHT.NUMBER(Ex 1 ) 2020/2744571-1/200 | LEGAL.RIGHT.REF | | DISB.AMOUNT | DISBURSE.AMT | |  | FULL.OR.PART | | BENEFICIARY>BEN.CUSTOMER | BENE.NAME | | BENEFICIARY@ID | BENE.NATION.ID | | BENEFICIARY>BEN.ACCT.NO | ACC.NO | | KID.NUMBER | KID.NO | |  | PAY.INFO | |  | INTERNAL.FLD | | DEPT.CODE>USER  DEPT.ACCT.OFFICER> NAME separated with “-” | ATTEST.BY | | SYSTEM DATE | DATE |  1. Form an OFS message to create a record in EB.HUS.ATST.DISBURSE table with the above field values in INAU status |
| Special Instructions | NA |

## HusValKidNumber

|  |  |
| --- | --- |
| Property | Specification |
| Type | S |
| Attached To | PAYMENT.ORDER,HUS.AMEND |
| Attached As | Validation Routine |
| Dependency | NA |
| Description | Routine to validate the kid number in the payment order. |
| Arguments – IN |  |
| Arguments – OUT |  |
| Prelim Conditions | NA |
| Subroutine Flow | When HUS.KID.NO field is amended from the Loan amendment screen, then HUS.KID.NO has to validated with Luhn Algorithm (MOD10 or MOD11) algorithm as described below,  1) Store the HUS.KID.NO in a local variable and set the data type as String.  2) Initiate a boolean variable and pass the HUS.KID.NO to IsMod10 method.  2) Initiate a method IsMod10 to find whether the provided number is validated as MOD10 using Luhn Algorithm.  3) Define a integer array such that the maximum length of the array is equal to the total length of the provided number.  4) Define a FOR loop where the counter begins with 0 and loops until the total length of the provided number. Increment the counter by 1 for every loop.  5) Store in a character variable with each character in the provided number by prefixing a space and assign it back to the integer array mentioned in step 3.  6) Define a FOR loop where the counter begins with integar array length minus 2 and loops until the counter is greater than or equal to 0. Decrement the counter by -2 for every loop.  7) For each loop, get the number from the integer array and multiply by 2. If the number is greater than 9, then initiate a variable and assign the calculated value where calculated value should be number modulo 10 plus number divided by 10. Example: number equal to 18 then 18%10 + 18/10 = 8 + 1 = 9  8) Assign the variable to the integer array. Follow the same until the loop completes.  9) Initialise a variable and sum all the numbers in the integer array.  10) If the sum is divisible by 10 i.e. sum%10 is equal to 0 then return true. If not, then return as False.  11) If the Boolean variable is set as False, then pass the HUS.KID.NO to IsMod11 method.  12) Initiate a method IsMod11 to find whether the provided number is validated as MOD11.  13) Define a integer array such that the maximum length of the array is equal to the total length of the provided number.  14) Define a FOR loop where the counter begins with the last digit of the number(total length) and loops until the first digit of the provided number. Decrement the counter by 1 for every loop.  15) Initiate a variable called multiplier and assign it as 2. For every loop, increment the multiplier with 1.  16) For each digit in the loop (step 14), multiple the digit with the multiplier and store it in a sum variable.  17) Repeat the loop until all the digits in the provided number is calculated and the sum variable will have the sum of all the multiplied values.  18) Calculate the modulo of the sum by 11 where remainder has to calculated and stored in a variable.  19) If the remainder is greater than 1 then subtract the remainder with 11 and if the resultant is the check digit.  20) If the remainder is 0 or 1, then add the reminder to the sum variable and store it in a variable X. Subtract the X with 11 and the resultant is the check digit. |
| Special Instructions | NA |

## 9.5 HusPpStatusActDisbUpd

|  |  |
| --- | --- |
| Property | Specification |
| Type | S |
| Attached To | PP.STATUS.ACTION |
| Attached As | Api Routine |
| Dependency | NA |
| Description | Routine to update the EB.HUS.DM.DELIVERY record for Letter of Disbursment |
| Arguments – IN |  |
| Arguments – OUT |  |
| Prelim Conditions | NA |
| Subroutine Flow | 1.Read .  2  4.End of the program. |
| Special Instructions | NA |

# Appendix

NA

# Accounting

NA

# Limits

NA

# COB Processing

NA

# Assumptions

NA

# Dependencies/Exclusions

This requirement is dependent on the following solutions:

|  |
| --- |
|  |
|  |

# Testing Notes

Please find DIT in the attachment.

# Packaging / Installation

Data Records and Source will be packed through dsf methodology

# Data Migration

NA