Card Management System (CMS)

Version 1.7.3 (3.5.1 RI0024.3)

Data Migration Approach

Document prepared by

FINANCIAL SOFTWARE & SYSTEMS (P) LTD
G 4, 1st Cross Street
SIPCOT IT Park
Rajiv Gandhi Salai (OMR)
Siruseri, Navalur
Chennai, TN – 603 103 India
Phone: +91-44-47415600
www.fss.co.in

Version 1.2

Restriction on Disclosure

The information contained in this document is confidential. This document shall not be disclosed, used or duplicated, in whole or in part, for any purpose other than to internally disseminate this information within FSS and InComm.

Document Control

Author	Sachin Nikam
Reviewed By	Shyamjith Palat
Reason of Change	Initial Draft
Date of Change	18-May-2012
Review Date	22-May-2012
Version Number	1.0

Ver. No	Author/s	Reason of Change	Date of Change	Reviewed by	Review Date
1.1	Dhiraj Gaikwad	Revised for the 1.7.2 (3.5.1 RI0024.2) release	10/June/2013	Abhijit K	17/June/2013
1.2	Sagar More	Includes cal logging features	15/Oct/2013	Abhijit K	15/Oct/2013

Table of Content

1 Synopsis

FSS Card Management System (CMS) supports migration of card data from the issuing institution's existing systems to ensure that cards already issued by the institution continue to work when the issuing institution goes live with CMS.

Data migration is typically a 'one-off' activity prior to go-live. Any ongoing data loads required on a frequent or ad-hoc basis are not part of the data migration scope as the migration program may require changes to match the upgraded product version and pre-requisites.

The purpose of this document is to define the approach to migrate information of existing customers, accounts, cards, and transactions into CMS version 3.5.1 RI0024.3.

2 CMS Data Migration

The CMS data migration process typically includes the following phases:

- **Planning**

  This involves creating the migration plan, aligning resources, setting milestones and finalizing a detailed execution plan.

- **Analysis and Design**

  This phase includes data profiling, validation of business requirements and rules, finalizing file formats, designing and customizing the migration routine and finalizing the migration strategy.

- **Mock Migration**

  In this phase, an end-to-end test run is executed to verify the migration procedure, validate the file formats and benchmark the migration time. Multiple cycles of mock migration can be executed depending on the complexity of the data. A cycle of mock migration can also be done on a subset of the production data in a pilot environment. This is followed by exception management and data integrity verification.

- **Live Migration**

  During this phase, Pre-migration checks are done and the full scale data migration is executed. Success and Exception Reports are generated for the activity.

- **Validation & Post-Migration Activities**

  After executing the migration process in the live environment, data validation is done. The reports generated during the migration process are reviewed and record count verifications are done. Variances and Exceptions are resolved or reconciled. Application is connected to the migrated data container and sanity is done. The system is monitored for a defined period and the migration process is signed off by stakeholders.

3 Definitions of Different Types of Data:

3.1 Master Data

Identified as fixed data, it describes the basic information that is involved in formation of major business entities like customer, account, card and call logging details.

These data types tend to be created once and maintained over a long time frame.

For Examples

1. For customer setup, country and state details are pre-requisite to migrate customer address.
2. For card setup, related BIN, product information and product parameters are pre-requisite.

3.2 Transaction Data

Transaction data describes business activities conducted by the institution. In the context of CMS, these transactions will be related to the card issuance and transaction authorization processes and related to:

- Existing Account information
- Existing Customer information
- Existing Card information
- Existing transactions done on the card
- Existing call log details with respect to transactions from call center

4 Data Migration Activities:

The complexity of data migration demands a process that ensures the accurate and complete transfer of data into the new system from the legacy system. The activities involved in the data migration process are detailed below:

4.1 Define the CMS Data Requirements (Functional)

In CMS, a Data Object is a business data unit such as customer. The CMS functional data requirements define the CMS data units that need to be migrated from the legacy system.

4.2 Define the CMS Data Requirements (Technical)

These define in detail the CMS Data structures, tables and fields. This includes the field name, field attributes (e.g. data type and length) and field properties (mandatory, optional, conditional or suppressed). This process requires a detailed knowledge of the associated CMS business processes and careful analysis of the configured system.

4.3 Determine the Data Transfer Method

For each business object the following choices are available for data transfer:

- To use the standard CMS data migration programs.
- To manually enter data using the UI available in CMS.

Data volumes, data complexity and availability of standard CMS data load programs are all to be considered before deciding between automated or manual load.

4.4 Design Data Migration Program

Where a data load process has been chosen as the best method to load specific data into CMS, the data load programs are designed and a technical specification written. The technical specification will define an appropriate CMS load technique and all data mapping (legacy field to CMS field).

4.5 Develop Data Migration Program

The data migration programs will be developed using the technical specification defined.

4.6 Manual Data Entry

Where manual data entry has been chosen as the best method to load some data into CMS, data will be manually entered into CMS using the appropriate UI options.

4.7 Identify the Legacy Data (Functional)

This activity identifies where the legacy data currently resides, in which applications/databases, and how it is currently entered and maintained. This activity is normally carried out by the legacy system owners.

4.8 Identify the Legacy Data (Technical)

This activity identifies in detail the approach in which legacy data can be extracted as per the CMS migration file specification. This process requires a detailed knowledge of the associated legacy applications and is normally carried out by the legacy system owners.

4.9 Legacy Data Cleansing

Legacy data required for migration into CMS must be completely cleansed prior to the final data load in CMS with the aim to ensure the consistency and accuracy of the data.

As a general approach the data will be cleansed on the legacy database before extraction.

Cleansing is an iterative process that can start as soon as the data has been identified as being required for migration into CMS. The data cleansing cycle includes the following steps:

- The elimination obsolete records.
- The removal of duplicate records.

4.10 Data Mapping and Transformation

This is a manual process where the data fields in the legacy data source are assigned corresponding fields in the CMS. Field text in the legacy system rarely agrees with the corresponding terminology in the CMS, therefore a variety of mapping methods are required. At the end of this step every CMS field that requires data must have been:

- Assigned a corresponding field from the legacy system.
- Assigned a transformation – i.e. converted from the original state to the required state using variety of methods including combining fields or logical rules for data transformation.
- Assigned a constant value.
- Generation of internal sequence numbers which may used for relating objects or for the formation of keys.

4.11 Identify Missing Data

Missing data is identified during the data mapping process, where a table or field in the CMS does not have an identifiable source from the legacy application.

4.12 Resolve Missing Data

CMS modules may require data that does not exist in the legacy systems. The approaches to resolve this issue are:

- Populate the missing data within the data migration programs, either by calculation or mapping tables.
- Developing guidelines for use during manual entry, which shows how to determine missing values while entering the data.

4.13Data Loading Instructions

Instructions for the data load will be written and agreed for all data objects in scope. This includes the order in which data is to be migrated and any dependencies.

4.14 Mock Migration - Execution

This is the process of running trial data load into the CMS which will help to ensure data accuracy, the correct load sequence, correct any loading errors and determine load duration. The trial data loads will continue until the load process completes successfully. We will process two cycles of trial data load or mock migration.

4.15 Mock Migration - Verification

After the mock migration execution is completed, the data loaded will be checked for the accuracy to ensure it meets defined requirements.

Exception report will be shared with bank after Mock migration 1. Post discussion on exception reports, rejected records can be migrated by changing the logic or by relaxing some criteria and this will be handled in Mock migration 2.

After Mock migration, InComm has to verify and confirm the migrated records. Post data check and confirmation final migration will be executed.

4.16 Execution Plan for Final Data Migration

This plan gives an indication of the feasibility of the final conversion into the production system in the available time frame, and specifies the data sequencing as well as the appropriate time to freeze the legacy system(s) for extracting information for the final migration.

4.17 Dual Data Maintenance

Master data will be migrated before transaction data migration. After the Master data migration is done, any changes in the Master data on the legacy system must also be reflected in CMS. Dual maintenance is best avoided if there is a large volume of master data changes as the additional workload may be excessive.

4.18 Data Cutover

This is the time for the final Data Migration into the CMS production system. The cutover period is the time between the shutting down of the old system (and ceasing of all logistic

related business activities) and commencement of use of the CMS. During cutover all processing on the legacy system is frozen and the data is extracted.

4.19 Extract Legacy Data

Migration data is extracted from the legacy system in the specified file format.

4.20 Data Migration

The data extracted from the Legacy system is read by the CMS data migration program and loaded into staging tables in CMS database. Extracted legacy data will be held on the data migration staging area for further analysis, manipulation and cleansing using the Migration utility. Integrity checks are required to ensure the correct data has been extracted. For example, all records are included in the extract; duplicates have not been created etc. After all integrity checks and validations, the CMS data migration utility will migrate this data into the various segments in CMS DB.

4.21 Reconcile the Migrated Data

This process checks that the data migrated into CMS meets the specified data requirements. This includes, but is not limited to manual data checks, record counts, checking balances, running reconciliation reports, approval of acceptable differences (rounding errors) etc. Success and Exception reports generated by the CMS migration utility are used for this reconciliation.

4.22 Data Migration Sign-off

After the migrated data has been reconciled and checked, the Data Migration Process will be signed-off. The sign-off will act as the approval for the go-live of the CMS application.

5 CMS Data Migration Methods

There are two primary methods of transferring data from a legacy system into CMS. The most appropriate method will depend on the volume and complexity of data to be migrated for each data object.

5.1 Migration Utility

This is a facility provided by CMS that enables migrated data to be first cleansed in staging, and subsequently loaded into CMS via predetermined field mapping. The facility also caters for 'data transformation' where field values can be created from simple predefined logic.

The CMS Migration utility is made up of the following steps:

- Read data (legacy data in flat files as per specified format).
- Migrate data (to the database used by the CMS application).

The following data units will be migrated using the CMS Data Migration utility:

1. Account
2. Customer
3. Card
4. Transactions
5. Call Log

5.2 Manual Migration

If the volume of data to be migrated is relatively low or is of poor quality, the overhead of developing migration programs might not be justified. Manually keying the data using the CMS transactions can enable the CMS logic to validate the data at the time of entry. The main disadvantages are the possibilities of input errors (mis-keying) and for multiple CMS rollouts the data must be manually entered each time.

The following data units will be migrated by manually keying in the information in the CMS application:

1. BIN
2. Product
3. Product Category
4. Product Profile/Parameters

5. Product Category Profile/Parameters
6. PIN and CVV keys
7. Fees
8. Product/Product Category Fee mapping
9. Limit profile
10. Product/Product Category Limit mapping
11. SMS and Email alert configuration at product category
12. Inventory Merchant Information
13. Inventory Location/Store Information
14. Host and CS Desktop Users

6 CMS Data Migration Process

6.1 Steps Involved

The complete migration process will be divided into following steps:

1. Master data setup
2. Validating input files and reading of data.
3. Loading of files data in to staging area
4. Pre-check on loaded data
5. Transformation of data in respective tables
6. Report generation
7. Post Migration checks

6.1.1 Master Data

Prior to actual data migration, master data will be populated into CMS. The master records are pre-requisite for migration of corresponding card/customer information and to support CMS functionality post-go-live.

The Master data for the following units should be manually keyed in so that corresponding accounts, customer, card and transaction data can be migrated using the migration utility program. These data units are listed in Section 4.2 above.

Note: For data migration, a user with name "MIGR" will be created in the CMS system. For all migrated records, the migration utility will associate this user as the inserted user and updated user for the corresponding data unit.

6.2 Reading and extraction of legacy data

For reading and extracting legacy data, the Oracle utility UTL_FILE will be used. Following are the entities for which pipe separated files will be generated from legacy system.

- **Account**

  This file will hold accounts details like number, status, type, balance etc.

- **Customer and card**

  This file will contain customer and card data.

- **Transaction**

  This file will hold complete transaction data for migrated cards.

- **Call Log**

  This file will hold call log details for transactions performed on particular card from call center

The detailed list of entities that should be part of each of the above files is listed in the CMS Migration File Specification Document.

File level validations

1. **File name**

   File name will be used to identify data unit included in the file.

   Format for the file name will be XXXX_NNNN.txt

   where:

   XXXX : Prefix including starting 4 characters of data unit. For Ex: for Account file it will be ACCO, for transaction it will be TRAN for customer it will be CUST, for call log it will be CALL.

   NNNN: File number. Max size of a file will be 500 Mb. If file size is more than 500 Mb, then additional files will be created.

   For Ex: if total size of Account file is 750Mb then 2 files will be created as:

   ACCO_0001.txt of size 500mb

   ACCO_0002.txt of size 250mb

List of Valid Prefixes

Sl. No.	Data Unit/Entity	Prefix
1	Account	ACCO
2	Customer and Card	CARD
4	Transaction	TRAN
5	Call Log	CALL

2. **File Header**

File header will be used to identify start of the file.

Format for the file header will be FH_XXXX_NNNN_NNNNNNNN

where:

FH: Constant denoting File header

XXXX : Prefix including starting 4 characters of data unit. For Ex: for Account file it will be ACCO.

NNNN: Running serial number of the file in case of multiple files left padded with zero

NNNNNNNN: Total number of records in file left padded with zeroes.

3. **File footer**

File footer will be used to identify end of file.

Format for the file footer will be FF_XXXX_NNNN_NNNNNNNN

where:

FF: Constant denoting File footer.

XXXX : Prefix including starting 4 characters of data unit. For Ex: for Account file it will be ACCO.

NNNN: Running serial number of the file in case of multiple files left padded with zero

NNNNNNNN: Total number of records in file left padded with zeroes.

4. **Number of records**

Maximum number of records allowed in a file will be 100,000.

5. **File size**

Maximum file size allowed will be 500 Mb. If file size is more than 500mb then file must be split into multiple files at the source system.

6.2.1 Data Cleansing

Before data can be successfully migrated data needs to be clean. Data cleansing is therefore an important element of any data migration activity.

- Data needs to be in a consistent, standardized and correctly formatted to allow successful migration into CMS.
- Data needs to be complete, to ensure that upon migration, all fields which are mandatory in CMS are populated. Any fields flagged as mandatory, which are left blank, will cause the migration to fail.
- Data needs to be de-duplicated and be of good quality to allow efficient and correct support of the defined business processes. Duplicate records can either be marked for deletion at source (preferred option), or should be excluded in the extract/conversion process.
- Legacy data fields could have been misused (holding information different from what this field was initially intended to be used for). Data cleansing should pick this up, and a decision needs to be made whether this data should be excluded (i.e. not migrated), or transferred into a more appropriate field.

Data Cleansing at source

It is the responsibility of the data owner (i.e. the business) to ensure the data provided for migration into CMS (File generated specifically for the CMS migration process) is accurate.
Data cleansing should, wherever possible, be done at source, i.e. in the legacy systems, for the following reasons:

- Unless a data change freeze is put in place, extracted datasets become out of date as soon as they have been extracted, due to updates taking place in the source system. When re-extracting the data at a later date to get the most recent updates, data cleansing actions will get overwritten. Therefore cleansing will have to be repeated each time a new dataset is extracted. In most cases, this is impractical and requires a large effort.
- Data cleansing is typically a business activity. Therefore, cleansing in the actual legacy system has the advantage that business people already have access to the legacy system, and are also familiar with the application. Something that is not the case when data is stored in staging areas.

- If data cleansing is done at source, each time a new (i.e. more recent) extract is taken, the results of the latest cleansing actions will automatically come across in the extract without additional effort.

Pre-Checks and Validations in CMS

Following are the pre-checks that will be applied on data which is loaded in staging area:

1. In all files, first check will be for mandatory data check as per the file format shared. If any mandatory field found as Null (Blank) then records will be marked as error.
2. Any record marked as error during validation will not be considered for Migration. Information for such records will be shared with bank as a part of exception report.

In addition to this, the following validations will be executed:

Validation for Account data:

1. Length of account should be 20
2. Account type should be either 01 or 02
3. Account status should be either 2 (Closed), 3 (Primary open), or 8 (Secondary open)

Validation for Card and customer data:

1. Title should be either 0 (Mr.), 1(Ms.), 2 (Mrs.) or 3 (Dr.)
2. State should be available in master
3. Country should be available in master
4. Email should contain @ symbol
5. Card number should be in clear form and should not be available in CMS
6. Current Card status should be available in master. For all cards, card status is mandatory.
7. Starter flag should be 0 or 1

8. KYC flag should be 0(ID FAILED) ,1(ID SUCCESS),2(IQ FAILED),3(IQ SUCCESS),4(KYC OVERRIDE),5(KYC PENDING)
9. Total accounts should be in between 1 (Min) and 6 (Max)
10. Account number should present in account file
11. Inventory Merchant ID should be available in master
12. Inventory Merchant Location ID should be available in master
13. Emboss gen flag should be either 0 or 1
14. Pin Gen flag should be either 0 or 1
15. All security questions and answers mandatory in case of GPR card
16. All mailing address details mandatory in case of GPR card
17. Proxy number length should be 9 or 12
18. Activation Date is mandatory for all cards having status other than Inactive
19. ID Type should be SSN (Social Security Number),DL(Driving License),PASS(Passport)
20. SSN length should be 10 in case of ID Type SSN.
21. Initial load flag should be either Y or N
22. SMS alert flag should be either 0 or 1
23. Email alert flag should be either 0 or 1
24. Product Code should be valid
25. Card Type should be valid

Validation for transaction log data:

1. Message type should be valid*
2. Delivery channel should be valid*
3. Transaction code should be valid*
4. Transaction type should be valid*
5. Transaction mode should be valid*

6. Response code should be valid*
7. Currency code should be available in master
8. Card number should be migrated in CMS
9. Proxy number should be migrated in CMS
10. Account number should be migrated in CMS
11. CR and DR flag should be valid
12. Reversal flag should be either 0 or 1
13. Narration is mandatory in case of financial transactions
14. Account and ledger balance before transaction and after transaction should be there.
15. Beneficiary card number is mandatory in case of Card to Card Transfer Transaction
16. Reversal code (00 – normal transaction). If it is other than 00 then it is a reversal
17. International indicator should be either 0 or 1 in case of ATM and POS transaction
18. Incremental indicator should be either 0 or 1
19. Completion count is mandatory in case of multiple preauth completions
20. Partial auth indicator should be either 0 or 1
21. Last completion indicator should L or Null (Blank)
22. Merchant floor limit indicator should be F,D,B,S, " " (Blank)
23. We are not doing any transaction processing.

Validation for Call Log data:

1. Call type should be valid*

Note Refer CMS Data migration file specification document for expected form of data.

6.2.2 Transformation of data in respective tables

Data mapping for Account information:

Account related information like Account number, Account status, Type, Account Creation Date, Balance will be mapped to objects which hold account related information.

List of tables impacted with table description:

Table name	Description
CMS_ACCT_MAST	This table holds account related information of customer.

Data mapping for Customer and Card information:

Customer and Card related information will be mapped to objects which holds customer and card related information. Customer and card related information will be like Customer name, permanent and communication address, email details, card number, status of a card, card activation and expiry date, ATM and POS transaction limits available, PIN offset, and linked accounts.

Note: For all cards emboss and PIN will be assumed as already generated.

List of tables impacted with table description:

Table name	Description
CMS_CUST_MAST	This table stores customer information
CMS_ADDR_MAST	This table stores address details of a customer.
CMS_CUST_ACCT	This table stores customer and account relationship
CMS_APPL_MAST	This table stores application information for generated cards.
CMS_APPL_DET	This table stores application code and account relationship
CMS_CAF_INFO_ENTRY	This table stores all application requests for generating cards.

Table name	Description
CMS_CUST_GROUP	This table stores customer group information.
CMS_PROD_CCC	This table stores product , product category and customer category relationship
CMS_SECURIRY_QUESTIONS	This table stores customer level security questions and answers
CMS_TRANSLIMIT_CHECK	This table stores card level ATM online/offline , POS online/offline, MMPOS online/offline transaction limits
CMS_MERINV_MERPAN	This table used to store information about card generated by inventory process.
CMS_MERINV_ORDR	This table stores merchant inventory order information
CMS_APPL_PAN	This table stores all card related information
CMS_PAN_ACCT	This table stores customer , account and card relationship
CMS_CARDISSUANCE_STATUS	This table stores application status of card
CMS_SMSANDEMAIL_ALERT	This table stores card level SMS and EMAIL alerts

Data mapping for transaction log information:

Transaction log information only for cards/accounts available in CMS will be migrated in CMS. Transaction information for all transactions done by card will be mapped in CMS. This information includes delivery channel, ISO message type, terminal id, transaction code and type, business date and time, total amount, merchant category code and ATM information.

Note: If card is not migrated in CMS and rejected by any reason, transactions for the same card will not be migrated in CMS

List of tables impacted with table description:

Table name	Description
CMS_STATEMENTS_LOG	This table stores successful transaction statement related information.

Table name	Description
CMS_TRANSACTION_LOG_DTL	This table stores all transaction related information.
TRANSACTIONLOG	This table stores all transaction related information.
CMS_MANUAL_ADJUSTMENT	This table stores manual adjustment transactions done from CSR channel.
CMS_DISPUTE_TXNS	This table stores dispute transaction details.
CMS_HTLST_REISU	This table stores information of cards which are replaced after hotlisting.
CMS_PAN_SPPRT	This table stores support function information happen on cards.
CMS_PREAUTH_TRANSACTION	This table stores preauth transaction information.
CMS_PREAUTH_TRANS_HIS	This is a history table for preauth and preauth completion transaction
CMS_C2CTXFR_TRANSACTION.	This table stores all card to card transfer request transaction done from CSR channel.

Data mapping for Call Log information:

Table name	Description
CMS_CALLLOG_MAST	This table stores call initiation log for every card number.
CMS_CALLLOG_DETAILS	This table stores the details of transactions performed on a single call.

6.2.3 Reports

The following reports will be generated during the Migration process:

Success Report

For successfully migrated records, summary and detailed report will be generated. In detailed report, major entities like card number, customer ID etc. will be shared with InComm.

Exception Report

Exception report will be generated for such records which are rejected due to some exceptions. For rejected records, summary and detail report will be generated. In detail report, major entities like card number, customer ID etc. along with migration error code and exception name will be shared with InComm.

Separate Reports will be available for data extraction and migration process. Please refer the CMS Data Migration Utility installation and execution manual for details of the reports.

6.2.4 Post Migration checks

Post migration data integrity checks will be carried out for the data which is successfully migrated.

1. Review and reconcile the success and exception reports generated.
2. Total number of successful records in staging tables should be tallied with total number of cards migrated.
3. Combination for card and account number given in input files should match with data actually migrated in the CMS segments.
4. Total number of transactions supplied in the input file should match with the transactions migrated into the CMS data segments.
5. InComm team will execute the PAN/Account control number update activity in which program will expect the product information and the serial number to be set for further issuance of cards and accounts from CMS. Please refer the CMS Data Migration installation and execution manual for detailed instructions on this.

6. Sanity should be done on the migrated data on a few random records by doing some routine business activities on the selected data through the CMS Host and CS Desktop applications.
7. Sanity should be done on the existing data present in the CMS DB before the data migration process by doing some routine business activities on such data through the CMS Host and CS Desktop applications.
8. Few test cards should be issued for all products and sanity should be done on such cards to ensure no impact on the card issuance and transaction processing modules due to the data migration activity.

7 Design Assumptions

1. Migration Utility will be executed directly from Database server.
2. Customer's Cardholder Website user id and Security question will be taken in clear format.
3. The Customer's Cardholder Website password and answers to security questions should be passed in hash SHA256 format and it is mandatory in case of GPR cards.
4. The migration process requires all 3 files as mandatory. The migration utility cannot be executed for individual files.
5. The PAN serial number generation logic for the product category should be configured as serial only.
6. The Inventory module in the CMS Host application should be used to configure Merchants and Locations that will be passed along with the starter cards sent in the card file.
7. Merchant and Location id fields will be mandatory in case of starter cards.
8. Transaction data will be migrated as it is without any processing logic for any transactions.
9. For PreAuth transactions preauth valid, expiry and completion flags are mandatory.
10. Pending hold amount will not be calculated. It will be accumulated based on the pending hold amounts received for each migrated preauth transaction.
11. There should be only one leg for card to card transfer transaction.
12. The migration utility will not do any special character validation on migrated data.
13. SMS and Email alerts configured at product category level and will be carried over to the migrated cards.
14. For starter card, if customer details are not available, dummy values should be provided for mandatory fields.
15. The migration process will need a complete downtime as the triggers on few tables will be disabled for the process.
16. A separate block will be provided to update PAN and Account serial numbers to be used for further card issuance. This block should be executed for every product and product category. Please refer the CMS Data Migration utility installation and execution manual for details on this process.

8 Roles and Responsibilities

The high level Roles and Responsibilities are provided below:

1. InComm will provide the input data to be loaded in the format specified by FSS. This responsibility will include the gathering and cleansing of source data. Please refer Migration file specification document for the file format details.
2. FSS has the responsibility for development of the data migration utility and testing it in the InComm development environment.
3. InComm will be responsible for executing all data migrations activities for QA, mock and production cycles.
4. InComm will validate data migrated in to CMS, and will sign-off formally at the end of the data migration cycles in QA, staging as well as production.

9 Key Issues & Risks

- There should be two Mock migrations followed by final migration on production environment.

  After mock migration 1, post verification of exception report for rejected records,

  - Data file shared may require correction of data if any and needs to be provided again with latest data.
  - Migration Logic may require change to migrate such records
- Changed logic/new files can be tested as Mock migration 2, but there are chances that other exceptions may be encountered during Mock migration 2. This will be due to unforeseen circumstances or due to some of the exceptions that were not encountered in Mock 1. For such cases, we may have to go for Mock 3.
- As the migration process will be executed on the existing CMS production database, a complete backup of the production database should be taken to restore in case restoration of original database is required at any time.
- The CMS data migration process is coded as per the specifications of the CMS version 3.5.1 RI0024.2 (InComm version 1.7.2). The migration program will need enhancement/changes to suit the version upgrades done in CMS at InComm.

10 References

Sl. No.	Document Name	Version	Dated
1	FSSTech_SD_CMS_Data Migration File Specification.pdf	1.7	15/Oct/2013
2	FSSTech_SD_CMS_Data Migration Installation and Execution Manual.pdf	1.5	15/Oct/2013