# ***CARDS – DOMAIN KNOWLEDGE***

## CARD NUMBER: -

* The Card Number is a **19 Digit Field**, Left Justified.
* The Card Number Identifies the CARD HOLDER.
* Card Number is also called **PAN** (Primary Account Number).
* The Number of Digits in the PAN is called the PAN-LENGTH (PANLEN).
* The Most common **PANLEN is 16** but we also see 15, 17, 18, 19.
* The Card Number is the **MOST IMPORTANT FIELD in determining where to ROUTE a TRANSACTION.**

## PREFIX: -

* The PREFIX is the **Leading Digits of a CARD NUMBER**.
* Prefix Length is usually **6 to 8 Digits**, but it can be more. (e.g., 9 or 10)
* **SWITCH Routing is based on PREFIX**, not based on entire CARD NUMBER.
* One of the Interesting **FUNCTIONS that the SWITCH must provide is Finding the PREFIX based on CARD NUMBER**.
* E.g., a FI (Financial Institution) may “own” the prefix 567123 with a PANLEN 18. This means they are free to ISSUE Cards like: 567123000000000001.

## MACCARD FILE: -

* The MACCARD File is a part of the Switch **COMMON DATABASE**.
* The **Primary Key** of this File is **PREFIX** Value.
* For Each PREFIX there is One Record in this file that the Switch knows about.
* There are **25,000** Records in MACCARD File in STAR NE Production Database.
* Each **MACCARD Entry** denotes a **PREFIX** for an **INSTITUTION that has a BUSINESS RELATIONSHIP with STAR NETWORK** and has **registered this PREFIX as belonging to that Institution.**
* A Financial Institution **(FI) can have MULTIPLE PREFIXES**.
* Important **FIELDS** in MACCARD File used for **ROUTING** –

1. PREFIX (must match leading digits of Card Number)
2. PANLEN (must match total number of Digits in Cardholder field)
3. PROCESSOR (where to send the Transaction)
4. ATM Only
5. POS Only
6. Signature Debit (SGD) Only
7. Electronic Benefits Transfer (EBT) Only
8. US-on-US Only.

## GATEWAY FILE: -

* **What is a GATEWAY**? A GATEWAY is a **Regional or Branded NETWORK (like STAR).**
* Is a GATEWAY an Acquirer or Issuer?
* Most Gateways CAN BE BOTH.
* How do we Connect to Gateways (as Authorizers)?
* A Gateway is like any other Processor. We Run a PI process that understands that Gateway’s format.
* How do we know what Transactions to Route to Gateways?
* The Gateway sends up a List of their PREFIXES (including Pan Length)
* Can a Financial Institution be a member of both STAR and another GATEWAY?
* YES

## SETTLEMENT: -

* SETTLEMENT is a process by which ASSOCIATIONS, NETWORKS and FI’s exchange Funds and Information.

## **CARD TRANSACTION PROCESSING – KEY PLAYERS: -**

### CARDHOLDER –

* Someone who obtains a Bankcard (Credit or Debit) from a Card Issuing Bank.

### ISSUER (CARDHOLDER BANK) –

* An Issuing Bank Issues Credit Cards to Consumers.
* ISSUER (Cardholder Bank) PAYS Acquiring Banks for Purchases that their Cardholders make.

### MERCHANT –

* A Merchant is any Business that Sells goods or services and maintains a MERCHANT ACCOUNT that enables them to Accept Credit or Debit Cards as PAYMENTS from Customers.

### PROCESSOR –

* The PROCESSOR acts as a MEDIATOR between Merchant and the Financial Institutions involved in Transactions.
* The Processor AUTHORIZE transactions and ensure Merchants gets Paid on time by facilitating the Transfer of Funds.

### ACQUIRER (MERCHANT’S BANK) –

* ACQUIRER CONTRACTS with Merchants to CREATE and Maintain Accounts (Called Merchants Account) that allow the Merchant to Accept Credit and Debit Cards.
* Acquirer also provide Merchants with EQUIPMENTS and SOFTWARE to accept Cards.
* The ACQUIRING BANK also deposits Funds from Credit card Sales into a Merchant Account.

### PAYMENT NETWORKS (VISA, MASTERCARD, STAR) –

* A PAYMENT NETWORK is the INTERMEDIARY between all the various Players in the Transaction.
* PAYMENT NETWORK provides CONNECTIVITY between the ISSUER and the MERCHANT.
* As an Intermediary, the PAYMENT NETWORK makes sure that regardless of whether the card is used at a Merchant or an ATM, the AUTHORIZATION and Funds movement are properly directed to ensure Money is appropriately distributed.

## **Transaction AUTHORIZATION at DEBIT SIDE** :

STEPS –

1. CARDHOLDER gives a NETWORK CARD to pay for Purchases.
2. MERCHANT swipes the CARD, enters the Amount and transmits an AUTHORIZATION Request to the MERCHANT BANK (ACQUIRER).

[ When Card is swiped in the POS/ATM device , it captures the Account information contained on the Card ]

[ Card Details are transmitted from POS to the MERCHANT ACQUIRER / MERCHANT ACQUIRER’S PROCESSOR ]

1. ACQUIRER (MERCHANT BANK) electronically sends the Authorization Request to NETWORK.

[ ACQUIRER PROCESSOR Identifies the payment NETWORK affiliated with the Card and forwards the details ]

1. NETWORK passes the Request to the ISSUER (Card Issuer Bank).

[ NETWORK receives the request for Authorization, Identifies the Card Issuing BANK and Routes the Transaction to the ISSUER PROCESSOR ]

1. ISSUER (Card Issuer Bank) Approves/Declines the Transaction and sends the RESPONSE back to the NETWORK.

[ ISSUER PROCESSOR receives the Request from NETWORK and executes a series of Inquiries (Risk of Fraud, Fund Availability etc.) ]

[ ISSUER PROCESSOR then Approves/Declines the transaction and sends back the response to the NETWORK ]

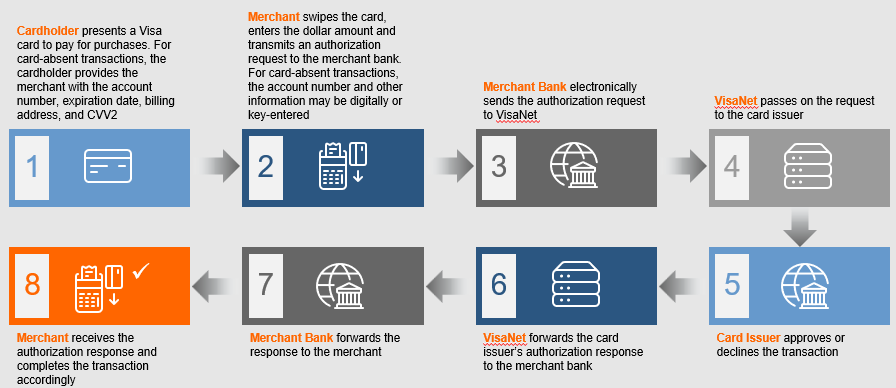
1. NETWORK forwards the AUTHORIZATION RESPONSE to the ACQUIRER (Merchant Bank).

[ NETWORK receives the Approval/Decline response and FORWARDS it to the ACQUIRER PROCESSOR ]

1. ACQUIRER (Merchant Bank) forwards the Response to the Merchant.

[ ACQUIRER PROCESSOR sends the Authorization Response back to the POS Device for Confirmation ]

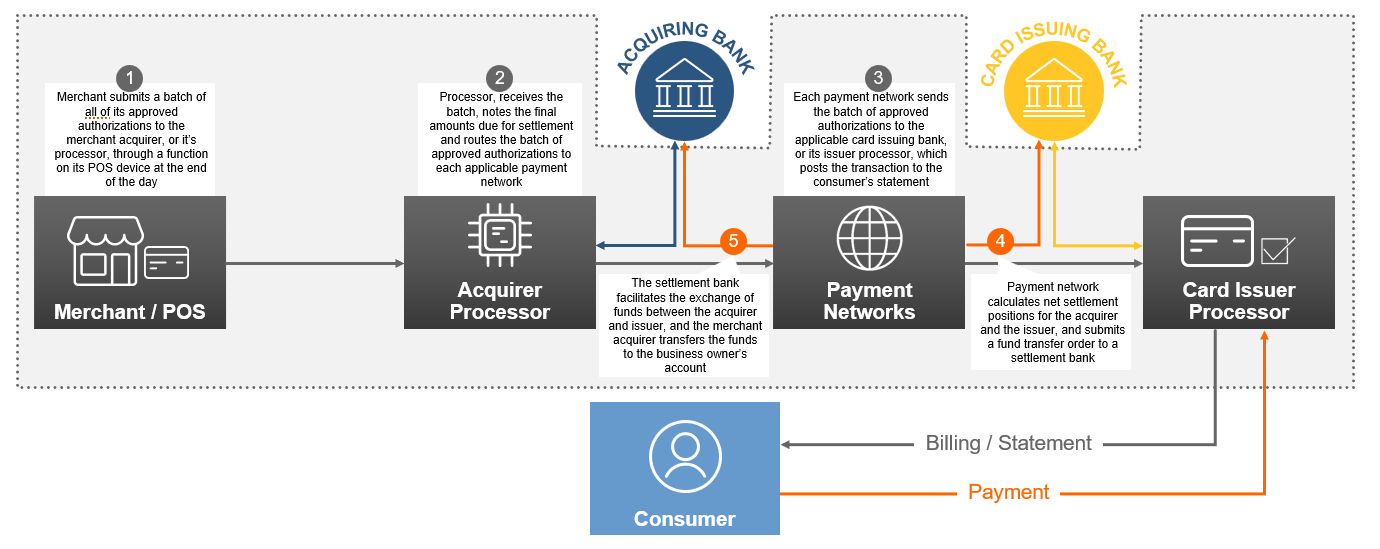
1. MERCHANT Receives the Authorization Response and completes the Transaction accordingly.



## **Transaction CLEARING and SETTLEMENT at DEBIT SIDE** :

STEPS –

1. MERCHANT submits BATCH of all Approved AUTHORIZATIONS to the MERCHANT ACQUIRER / ACQUIRER PROCESSOR at the END OF THE DAY.
2. ACQUIRER PROCESSOR receives the BATCH, notes the Final AMOUNTS due for Settlement and ROUTES the Authorizations to each Applicable NETWORK.
3. Each PAYMENT NETWORK forwards the Batch of Authorizations to the applicable Card ISSUING BANKS (ISSUERS), which Sends out the TRANSACTION to the CUSTOMER’S STATEMENT.
4. NETWORK then Calculates NET SETTLEMENT positions for ACQUIRER and ISSUER and submits a FUND TRANSFER ORDER to the SETTLEMENT BANK (ISSUER Bank).
5. The SETTLEMENT BANK (ISSUER Bank) facilitates the Exchange of Funds between Acquirer and Issuer and then the MERCHANT ACQUIRER (ACQUIRER Bank) transfers the Funds to the MERCHANT (Business Owner) Account.



# ***ISO 8583 – MESSAGE TYPES & TRANSACTION FLOW***

QUES ----

1. Diff b/w CVV & ICVV?
2. How to identify Purchase and withdrawal transaction?
3. Processing Codes in DE3?

## **MESSAGE STRUCTURE**: -

* A Message Structure has 4 components-

1. MESSAGE HEADER
2. MESSAGE TYPE IDENTIFIER (MTI)
3. BITMAP
4. DATA FIELDS

### BITMAPS –

( BITMAPS serves as “TABLE OF CONTENTS” for the Message )

( We can look at the BITMAP and Decide which FIELDS are PRESENT based on each field’s format )

( BITMAPS are usually in HEX Format ; to see which Fields are present, we need to convert the Bitmaps from HEX to Binary format )

* BITMAP 1 -> Fields 2 – 64
* BITMAP 2 -> Fields 66 – 128
* BITMAP 3 -> Fields 130 – 192

### DATA FIELDS (DATA ELEMENTS) –

* DE 2 -> Primary Account Number (PAN) (16 Digits)
* DE 3 -> Processing Code / PROC Code (3 Digits)
* DE 4 -> Transaction Amount (6 Digits)
* DE 5 -> Settlement Amount (6 Digits)
* DE 6 -> Cardholder Billing Amount
* DE 9 ->
* DE 10 ->
* DE 11 ->
* DE 14 -> Expiry Date
* DE 15 -> Settlement Date
* DE 19 -> Acquirer Institution Country Code
* DE 22 -> PAN Entry mode
* DE 23 -> Card Sequence Number
* DE 28 -> Transaction Fees
* DE 32 -> Acquirer Institution ID
* DE 33 ->
* DE 34 -> PAN Extended
* DE 35 -> Track 2 Data
* DE 37 ->
* DE 39 -> Response Code

[ 00 -> Approved

04 -> Declined (Capture Card)

12 -> Declined (Invalid Transaction)

13 -> Declined (Invalid Amount)

14 -> Declined (Invalid Card Number)

30 -> Declined (Format Error)

51 -> Declined (Insufficient Funds)

55 -> Declined (Invalid PIN)

91 -> Declined (Issuer Inoperative)

92 -> Declined (Routing Not Available) ]

* DE 45 -> Track 1 Data
* DE 49 ->
* DE 51 ->
* DE 52 -> PIN Data
* DE 54 -> Additional Amounts
* DE 55 ->
* DE 56 -> Payment Account Reference Data
* DE 60 ->
* DE 63.1 -> Network Identification Code
* DE 66 -> Settlement Code
* DE 126.1 -> Customer Name
* DE 126.2 -> Customer Address
* De 126.9 -> CAVV Data

### MESSAGE SEGGREGATION INTO DIFFERENT COMPONENTS –

* Message Example –

**16010200e698151600000002100042c810290202ed000200723e648128e09826104227979999999999004000000000001591120421243366818123235612040911120553100840901000063192722504227971174405176d09111010000000877000f7f3f3f9f0f2f6f6f8f1f8f1f0f7f3f6f0f0f7f8f0f9f1f0f0f0f0f7f3f6f940404040e3c1d9c7c5e340e3f0f7f3f640d5d6d9c3d640404040404040d5969983964040404040404040e4e208402be84a8847a449a1200101010000000002f0f619400018000000000003873392667314702000050000f2f9f0f0098020200003e9c1f2f0**

* The First part is HEADER. (**16010200e698151600000002100042c810290202ed00)**
* The Second part is MTI (Message Type Identifier)

(**0200)**

* The Third part is BITMAPS.

( **723e648128e09826** ) -> HEX Format

Convert this Hex format into Binary –

* Result - 0111 0010 0011 1110 0110 0100 1000 0001 0010 1000 1110 0000 1001 1000 0010 0110

[ Starting from Left – Every “1” indicates that at that particular position the Field is present.

e.g., In the above Scenario for the First 8 Data Fields … Field 2, 3, 4, 7 field is only Present ]

* The Fourth part is the DATA FIELDS.

## **MTI (MESSAGE TYPE IDENTIFIER) STRUCTURE**: -

* MTI is a 4 Digit Numeric Character field. (xxxx)
* First Position - VERSION NUMBER

1. -> ISO 8583 (1987 Version)
2. -> ISO 8583 (1993 Version)

* Second Position - MESSAGE CLASSES

1 -> Authorization

2 -> Financial

3 -> File Action

4 -> Reversal

6 -> Administrative

8 -> Network Management

* Third Position - MESSAGE FUNCTION

0 -> Request

1 -> Request Response

2 -> Advice

3 -> Advice Response

* Fourth Position - TRANSACTION ORIGINATOR

0 -> Acquirer

1 -> Acquirer Repeat

4 -> Others

## **TRANSACTION TYPES**: -

* 1100 -> Authorization Request (Acquirer to Issuer)
* 1110 -> Authorization Request Response
* 1120 -> Authorization Advice
* 1121 -> Authorization Advice Repeat
* 1130 -> Authorization Advice Response
* 1200 -> Financial Request (Acquirer to Issuer)
* 1210 -> Financial Request Response (Issuer to Acquirer)
* 1220 -> Financial Advice
* 1221 -> Financial Advice Repeat
* 1230 -> Financial Advice Response
* 1304 -> File Action Request
* 1314 -> File Action Response
* 1420 -> Reversal Advice (Network to Issuer)
* 1421 -> Reversal Advice Repeat
* 1430 -> Reversal Advice Response (Issuer to Network)
* 1624 -> Administrative Advice
* 1634 -> Administrative Advice Response
* 1804 -> Network Management Request
* 1814 -> Network Management Request Response
* SMS – Single Message System
* DMS – Dual Message System

## **TRANSACTION FLOWS** –

### Financial Transaction Request (0200) and Financial Transaction Request Response (0210) [STANDARD MESSAGE FLOW]

STEPS ---

1. Acquirer initiates a Financial Transaction Request (200) to the Network.
2. Network sends the (200) message to the Issuer.
3. Issuer generates Financial Transaction request Response (210) and sends it to Network.
4. Network forwards the (210) message to the Acquirer.

### Financial Transaction Request (200) and Financial Transaction Request Response (210): PARTIAL APPROVALS with CURRENCY CONVERSIONS ASSESSMENT.

[This Scenario is only valid when DE111 (Amount, Currency Conversion Assessment) is Present and Acquirer and Issuer Currency is DIFFERENT. Not Valid for all Partial Approval Transaction]

STEPS ---

1. Acquirer initiates a Financial Transaction request (200) to the Network.
2. Network forwards (200) message to the Issuer.
3. Issuer generates Financial Transaction Request Response (210) and sends it to Network.
4. Network forwards the (210) message to the Acquirer.
5. If the Issuer provides a PARTIAL APPROVAL RESPONSE (DE39 = 10) in the (210) message, the Single Message System (recalculates currency and conversion amounts based on Partial Approval Amount).

and SMS / NETWORK sends revised Information to the ISSUER in the Financial Transaction Advice (220) message.

NOTE - If Currency conversion Assessment is recalculated and the amount is ZERO, then Financial Transaction Advice (220) message is not sent.

1. ISSUER sends a Financial Transaction Advice Response (230) message to the SMS / NETWORK.

### Financial Transaction Request (0200) and Financial Transaction Request Response (0210): DEBIT MASTERCARD (POS) PARTIAL APPROVALS

STEPS –

### Financial Transaction Request (0200) : DENIED BY NETWORK

STEPS ---

1. ACQUIRER initiates a Financial Transaction Request (200) to the NETWORK.
2. The NETWORK generates a Financial Transaction Request Response (210) to the ACQUIRER indicating Request Denial. 210 messages contain RESPONSE CODE (DE39) indicating reason of Denial.

### Financial Transaction (02xx) : POS PREAUTHORIZATION AND COMPLETION

STEPS ---

1. .

### Financial Transaction (02xx) : EXCEPTION, TRANSACTION FAILURE AFTER POS PRE-AUTHORIZATION

STEPS ---

1. .

### Financial Transaction (02xx) and Acquirer Reversal (04xx): EXCEPTION, TRANSACTION FAILURE AFTER POS PRE-AUTHORIZATION

STEPS ---

1. .

### Financial Transaction (02xx): EXCEPTION, NETWORK STAND-IN PROCESSING (Late Response from Issuer)

STEPS ---

1. ACQUIRER Initiates a Financial Transaction Request (0200) message to the NETWORK.
2. NETWORK forwards (0200) message to the ISSUER.
3. NETWORK detects a Time-Out condition on the Financial Transaction Request Response (0210) message that was expected from the ISSUER.

NETWORK creates an ACQUIRER Reversal Advice (0420) message that has DE60 (Advice Reason Code) with value 4020090 (timeout) indicating NO (0210) message was received.

NETWORK stores this (0420) Message in the SAF File for Later delivery to the ISSUER.

1. If ISSUER Processor is configured for STAND-IN processing, the NETWORK creates an INTERNAL Financial Transaction Request (0200) message and sends it to STAND-IN processing for AUTHORIZATION.

The STAND-IN Processing VALIDATES the request and creates an INTERNAL RESPONSE MESSAGE.

1. The NETWORK uses the Internal Response to create a Financial Transaction Request Response (0210) message and sends it to ACQUIRER.
2. The NETWORK creates a Financial Transaction Advice (0220) Message and places it in SAF FILE for Later Delivery to the ISSUER.
3. The ISSUER sends a LATE Financial Transaction Request Response (0210) Message to the NETWORK.
4. The NETWORK responds to the ISSUER with an Acquirer Reversal Advice (0420) Message containing DE60 (Advice Reason Code) with Value 4000000 (Late Response from Issuer).

(This Indicates to the ISSUER that the 0210 message is LATE and was REJECTED)

NOTE: - If Late Financial Transaction Response (0210) has a Response Code indicating a Request Denial, then NETWORK will NOT Take action i.e., 0420 Message is Not sent.

1. The ISSUER responds with a Acquirer Reversal Advice Response (0430) Message to the NETWORK.
2. The NETWORK Initiates a Network Management Request (0800) Message "ECHO TEST" to Verify/Establish communication with the ISSUER.
3. The ISSUER responds with a Network Management request Response (0810) Message.
4. When COMMUNICATION is established b/w ISSUER & NETWORK, the NETWORK sends a Acquirer Reversal Advice (0420) Message that contains DE60 (Advice Reason Code) with value 4020090 (IPS Timeout)
5. The ISSUER responds with a Reversal Advice Response (0430) message to the NETWORK.
6. The NETWORK sends a Financial Transaction Advice (0220) message with DE38 (Authorization Identification Response) to the ISSUER from SAF Facility. (which was earlier Stored in SAF File)
7. The ISSUER responds with a Financial Transaction Advice Response (0230) Message.
8. Any Remaining messages if present and stored in the SAF FILE for the ISSUER will be sent by NETWORK. NETWORK will then send a Network Management Advice (0820) message to the ISSUER indicating EOF for SAF FILE.

### Financial Transaction (02xx): EXCEPTION, NETWORK STAND-IN PROCESSING (NO RESPONSE FROM ISSUER).

STEPS ---

1. ACQUIRER Initiates a Financial Transaction Request (0200) message to the NETWORK.
2. NETWORK forwards (0200) message to the ISSUER.
3. NETWORK detects a Time-Out condition on the Financial Transaction Request Response (0210) message that was expected from the ISSUER.

NETWORK creates an ACQUIRER Reversal Advice (0420) message that has DE60 (Advice Reason Code) with value 4020090 (timeout) indicating NO (0210) message was received.

NETWORK stores this (0420) Message in the SAF File for Later delivery to the ISSUER.

1. If ISSUER Processor is configured for STAND-IN processing, the NETWORK creates an INTERNAL Financial Transaction Request (0200) message and sends it to STAND-IN processing for AUTHORIZATION.

The STAND-IN Processing VALIDATES the request and creates an INTERNAL RESPONSE MESSAGE.

1. The NETWORK uses the Internal Response to create a Financial Transaction Request Response (0210) message and sends it to ACQUIRER.
2. The NETWORK Initiates a Network Management Request (0800) Message "ECHO TEST" to Verify/Establish communication with the ISSUER.
3. The ISSUER responds with a Network Management request Response (0810) Message.
4. When COMMUNICATION is established b/w ISSUER & NETWORK, the NETWORK sends a Acquirer Reversal Advice (0420) Message that contains DE60 (Advice Reason Code) with value 4020090 (IPS Timeout)
5. The ISSUER responds with a Reversal Advice Response (0430) message to the NETWORK.
6. The NETWORK sends a Financial Transaction Advice (0220) message with DE38 (Authorization Identification Response) to the ISSUER from SAF Facility. (which was earlier Stored in SAF File)
7. The ISSUER responds with a Financial Transaction Advice Response (0230) Message.
8. Any Remaining messages if present and stored in the SAF FILE for the ISSUER will be sent by NETWORK. NETWORK will then send a Network Management Advice (0820) message to the ISSUER indicating EOF for SAF FILE.

### Financial Transaction (02xx): XCEPTION, SYSTEM FAILURE DURING ACQUIRER FINANCIAL TRANSACTION REQUEST (0200)

STEPS ---

1. .

### Financial Transaction (02xx): EXCEPTION, STAND-IN MAESTRO PREAUTHORIZATION.

STEPS ---

1. .

# ***FUP (FILE UTILITY PROGRAM)***

* FUP is a UTILITY designed to help you MANAGE DISK FILES and PROCESSES on a Non-Stop System.
* We can use FUP to CREATE, DISPLAY, DUPLICATE Files, LOAD DATA into Files, ALTER File Characteristics and PURGE Files.
* FUP Supports these types of ENSCRIBE Disk files –

1. Key Sequenced
2. Entry Sequenced
3. Relative
4. Unstructured (Tedit Files)

* FUP has a **Limitation that Command Line cannot exceed 132 Characters**.
* FUP is a component of the Standard Software Package for the TANDEM Non-Stop Kernel.
* FUP is used for FILE MANAGEMENT.
* To **TERMINATE a FUP Process** Once it has Started use – “**CTRL-Y**”.
* **ENSCRIBE** Files are CREATED using FUP.
* FUP reads 2 FILES (**FUPLOCL and FUPCSTM**) before it issues its Frist prompt. Both these files are Standard FUP COMMAND Files that contain ASCII text with valid FUP Commands.
* You can use only 1 FUP COMMAND as a TACL Command at a time in the TACL session.
* We can access FUP through TACL acting as a COMMAND INTERFACE.
* **WILD-CARD OPTION** 🡪 In FUP, we can use “**\***” or “**?**” as Wild-Card Characters to help specify Files.

e.g., FILEINFO $DAT040.COBOL.A\*DAT

## HOW TO START A FUP PROCESS –

* There are 3 ways to START a FUP Process –

1. ENTER FUP (at the TACL prompt) and then use FUP commands interactively.

e.g.,

> FUP

> DUP <filename1>, <filename2>

1. Write FULL COMMAND including FUP at the TACL Prompt and Enter.

e.g.,

> FUP DUP <filename1>, <filename2>

1. CREATE a FUP COMMAND FILE that contains FUP Commands.

* Create a TEDIT File.
* Write FUP Commands you want to execute (do not include FUP at the start of the command)

e.g.,

> SUBVOLS <subvolname>

> DUP <filename1>, <filename2>

* RUN the Command File on TACL Prompt.

**FUP / IN <commandfile> OUT <outfilename> /**

[ NOTE – You can add comments in command file; (--) to add comments ]

## **FILE CODES** –

1. 101 -> Tedit File / Unstructured File
2. 100 -> TAL Object File.
3. 800 -> PTAL Object File.
4. 1300 -> Audit Files
5. 0 -> Database Files.
6. 1729 -> Pak Files.

## **FILE FORMATS** –

1. **FORMAT 1 File** –

* Size -> Less than 2 GB.
* Max Block Size -> 4096 Bytes.
* Max Record Size -> [Block Size – 34 Bytes] (Key Sequenced)

[Block Size - 24 Bytes] (Entry Sequenced & Relative)

* Max Partition Size -> 2 GB – 1 KB.

1. **FORMAT 2 File** –

* Size -> Greater than 2 GB.
* Max Block Size -> 4096 Bytes
* Max record Size -> [Block Size – 56 Bytes] (Key Sequenced)

[Block Size - 48 Bytes] (Relative)

[Block Size - 44 Bytes] (Entry Sequenced)

* Max Partition Size -> 1024 GB

## **FILE ATTRIBUTES (ENSCRIBE FILE PARAMETERS )**—

1. MAXEXTENTS (Maximum Extents) –
2. TYPE (File Organization)
3. FORMAT
4. CODE
5. EXTENTS
6. BLOCK (Length of Block)
7. REC (Record Length)

## **FILE ATTRIBUTES / PARAMETERS that can be ALTERED** –

1. **MAXEXTENTS**

[ FUP ALTER <filename>, <PARAMETER> <VALUE>

1. **CODE** [101 -> Tedit ; 100 -> TAL Object ; 0 -> Database File ]

[129 -> Spooler Job File ; 131 -> Inspect File ; 200 -> DDL File ]

[ 800 -> PTAL Object ]

[ FUP ALTER <filename>, <PARAMETER> <VALUE> ]

## **FUP COMMANDS** –

1. **FUP INFO <filename>, DETAIL**

(This command will Display all the File Details like – File Format, File Code, Extents, Owner, Security, EOF)

1. **FUP $\*.\*.\***

( This Command will show all the Files available in all the Volumes and Sob-Volumes)

1. **FUP INFO <d?rec> [ WILDCARD OPTION]**

( This Command will Search & Display all the Files Starting with Character “D”, whose Second Character is not known but ends with “REC”)

1. **FUP DUP <filepath1>, <filepath2>**

( This command will Create a Duplicate file with the same Name & having the Same file Properties as the Original File; only when <filepath2> was not present earlier. )

1. **FUP DUP <$AB.CDE.\*>, <$PQ.XYZ.\*>**

( This command will Duplicate ALL the Files present in “$AB.CDE” Subvolume to “$PQ.XYZ” subvolume having the same data & same File properties as present in the Original )

1. **FUP DUP <filepath1>, <$vol.subvol.\*>, PURGE [FORCE COPY]**

( If the File exists in the Destination sub-volume “$vol.subvol” then it will Delete it first and then will Duplicate the file at the Destination Location )

1. **FUP INFO <filepath>**

( This command will Display Basic Information about a File like – Size, EOF, Code )

1. **FILEINFO <filepath>**

( This command Displays Basic Information about a File)

1. **FUP INFO <filepath>, STATISTICS**

( This command Displays the NUMBER of RECORDS in a File along with all the File Details ) ( command mostly used in case of DATABASE Files)

1. **FUP COPY <filepath1>, <filepath2>**

( This command will COPY File RECORDS from Filepath1 to Filepath2 only when File (Filepath2) is already Present at the location. )

1. **FUP COPY <filepath1>, <filepath2>, COUNT <N>**

( This command will COPY First “N” records from Filepath1 to Filepath2, provided that filepath2 is already present at the Location )

1. **FUP INFO <filepath>, EXTENTS**
2. **FUP DUP <filepath1>, <filepath2>, SOURCEDATE**

( This command DUPLICATES filepath1 to filepath2, Retaining the Original Date and Time of Filepath1 to Filepath2 )

1. **FUP LISTOPENS <filename>**

( The PROCESSES that are using this file can be Listed using this command ) ( If the File is in OPEN State, then this command will Display all the PROCESSES that have Opened this File )

1. **FUP RENAME <filename1>, <filename2>**

( This command is used to RENAME file i.e., filename1 to filename2 ) ( can Rename in the same Volume/sub-volume also )

1. **FUP COPY <filepath>,, A, SHARE**

( This command Displays the Record of an OPEN file and Displays it on the Terminal where, A -> ASCII ; B -> Binary ; H -> Hex )

1. **FUP PURGEDATA <filepath>**

( This command DELETES all the Records / Data from the specified filepath )

1. **FUP PURGE <filepath>**

( This command PURGES / DELETES the File )

1. **FUP LOAD <filepath1>, <filepath2>**

( This Command LOADS Data into the MAIN File only and Not in the Alternate-Key Files. The Data in the file being Loaded is OVERWRITTEN )

1. **FUP CREATE <filename>**

( This command CREATES a Disk File with the Current / Default File-Creation Attributes defined by the SET command ) ( By Default UNSTRUCTURED File [101] is created using this command )

1. **!**

( This Command Executes the LAST COMMAND you entered again )

1. **FUP ALTER <filename>, <PARAMETER> <VALUE>**

( This command Alters Characteristics / Parameters of an Enscribe Disk File )

( NOTE – To Alter a File, you must have both i.e., Read and Write access to it )

e.g.,

* FUP ALTER <filename>, CODE 101
* FUP ALTER <filename>, MEXEXTENTS 100
* FUP ALTER <file>, DELALTKEY “ab”, DELALTFILE 0

( To Delete Alternate Key “ab” from a File )

* FUP ALTER <filename>, ALTFILE (2, <altfilename>)

( command to Assign alternate Key File <altfilename> to <filename> and give it Key Number 2 )

1. **FUP CREATE <filename>, <PARAMETER VALUE>, …… <PARAMETER VALUE>**

( This command CREATES a Disk FILE with the Attributes defined in the Command )

e.g.,

* To Create UNSTRUCTURED File –

**FUP CREATE <filename>, TYPE U, CODE 101, FORMAT 1, EXT(250, 250), REC <N>, MAXEXTENTS 978**

* To Create KEY-SEQUENCED File –

**FUP CREATE <filename>, TYPE K, FORMAT 1/2, REC <N>, KEYLEN <N>, KEYOFF 0, MEXEXTENTS <N>**

1. **FUP GIVE <filename>, <USER-ID>**

( This command CHANGES the OWNERSHIP of the File. Command only applies to Enscribe Files )

( Only the Current Owner of the File can execute the GIVE command or the SUPER-ID (255, 255) )

1. **FUP COPY <filepath>,, A, SHARE, COUNT <N>**

( This command is used to Display / Read a Certain No. of Records (N) in an OPEN File & Display it on the Terminal, where –

A -> ASCII; B -> Binary; H -> Hex )

1. **USER <USER-ID>**

( To get the NAME of the PERSON from the User-ID )

1. **FUP LISTLOCKS <filepath>**

( Command to Check Whether any RECORD is LOCKED or NOT on the Present File )

1. **FUP SUBVOLS**

( command Displays ALL the SUBVOLUMES present in the current Volume )

1. **FUP FILES**

( command Displays ALL the FILES present in the current sub-volume )

1. **FUP DUP <filepath1>, <filepath2>, SAVEALL**

( command DUPLICATES the <filepath2> same as <filepath1> and INCLUDE OWNERSHIP and SECURITY Info same as the Source file <filepath1> )

1. **HISTORY <N>**

( Command Displays the FUP commands used Recently)

1. **SYSTEM <\node.$volume.subvol>**

( Command lets you SET Volume and Sub-volume as the DEFAULT VOLUME )

1. **STATUS $<process-name>**

( Command displays the Current STATUS of a particular PROCESS )

1. **FUP VOLS**

( Command displays all the VOLUMES in the current System )

1. **FUP ALLOCATE <filename>, <value>**

( Command ALLOCATES file EXTENT to a File. It applies to Enscribe Files Only )

( It is the Total Number of EXTENTS allocated to a File )

1. **FUP COPY $DAT040.COB.\* WHERE MODTIME >/< <DATE>**

( Command to COPY all the Files that have changed since a Specified DATE)

1. **FUP COPY <filepath1>, <filepath2>, FIRST <N>, RECIN <N>**

( Command to COPY First N records from filepath1 to Filepath2 and display First n Characters of each Record )

1. **FUP DEALLOCATE <FILENAME>**

( Command to DEALLOCATE the UNUSED EXTENTS from a File)

1. **FC**

( Command to MODIFY and REEXECUTE the Last Command entered )

1. **FILES**

( Command to DISPLAY all the FILES in the current Sub-volume )

1. **FUP INFO <filename>, EXTENTS**

( Command to DISPLAY the Details of Extents Allocated to the File )

1. **FUP** **OBEY <filename>**

( This Command READS the Commands present inside the File and EXECUTES them)

## **Difference Between FUP COPY / DUP / LOAD** :-

* FUP LOAD 🡪
* This command LOADS Data only in the MAIN FILE (Not in the Alternate-Key Files).
* DATA in the file being Loaded is OVERWRITTEN.
* To LOAD Data in Alternate-Key Files we use Command –

FUP LOADALTFILE <filename>

* FUP LOAD is Much FASTER than FUP COPY.
* FUP DUP 🡪
* This Command DUPLICATES files along with their DATA.
* FUP DUP Creates IDENTICAL Copies of Files (i.e., the File Attributes also remain Same as Original)
* Destination / Duplicated File should not be present beforehand performing FUP DUP else it will throw an Error.
* FUP COPY 🡪
* This Command COPIES Data / Records from Source File to Destination File.
* Destination File should be Present beforehand performing FUP COPY else it will throw an Error.

## **CREATE a PARTIONED ALTERNATE KEY-SEQUENCED FILE** :-

STEPS –

* FUP
* SET TYPE K
* SET CODE 101
* SET EXT (32, 32)
* SET REC 80
* SET BLOCK 4096
* SET KEYLEN 2
* SET ALTKEY (“LO”, KEYOFF 10, KEYLEN 4)
* SET ALTKEY (“VN”, KEYOFF 14, KEYLEN 6)
* SET ALTFILE (0, <alt-filename>)
* SET PART (1, $ade001, 5, 5)
* CREATE <FILENAME>

# ***TAL (TRANSACTION APPLICATION LANGUAGE)***

* The TAL (Transaction Application Language) is a HIGH-LEVEL, BLOCK STRUCTURED Language that works efficiently with the system hardware.

# ***SWITCH***

* SWITCH acts as an INTERFACE between ACQUIRER and ISSUER. It Directly controls and Manages ATM Devices and Transactions.
* SWITCH ROUTES Transactions from ACQUIRER to ISSUER.

## **TRANSACTION FLOW** –

Acquirer Router Issuer Router

Attala PMC ) Issuing PI

pend Acquiring PI (Processor Interface) DCH

logger Issur Bank

DCH (Data communication handler)

Terminal Handler 🡪(ATM/POS/Self Point)

* DCH 🡪 It REMOVES all the HEADER from the Message and Convert Native Form Message to PI Understandable Form.
* PI 🡪 It Sends Transaction to AUTHORIZER (ISSUER) and Matches the Replies (From Issuer) From Original Transaction Request.
* ACQUIRING PI 🡪 It VALIDATES the Message and Convert ISO to IPC (Request Side) and IPC to ISO (Reply Side)
* ISSUING PI 🡪 It Converts IPC to ISO (Request) and ISO to IPC (Reply).
* PMC (Primary Message Controller) 🡪 It is used for MESSAGE / CARD PREFIX VALIDATION. It also Provides PIN Key Management Services via ATTALA Hardware Encryption Device.
* ATTALA 🡪 It is a SECURE BOX, mainly used for PIN Validation Decryption / Re-Encryption.
* ROUTER 🡪 It ROUTES the Transaction between ACQUIRER and ISSUER and also Reduces the Communication Overloads.
* PMC 🡪 Primary Message Controller. It performs the following Functions –
* Determines ROUTING for the Transactions.
* Verifies ELIGIBILITY of Cards for Specific Products.
* Verifies ELIGIBILITY of Cards for Specific Transaction Types.
* Performs CVC VALIDATION.
* LOGGER 🡪 Logger is a PROCESS which LOGS all Types of SWITCH Transactions and NETWORK MESSAGES.
* PEND 🡪 Pend is a FILE that STORES all APPROVED FINANCIAL Transactions.
* NETCON 🡪 Network Console Process. Netcon Processes Input COMMANDS and passes them to appropriate PROCESS, sends Switch STATUS Messages to Network Console and Logger.

## **NETWORK FORMATTERS** –

### STAR Network Proprietary PI Formatters :

1. PIESTRD [ STAR ISO - NE Platform ]
2. PIEPSD [ MAC ISO - NE Only ]
3. PISTARD [ STAR ISO - SE / West Platforms ]
4. PIMASM3D [ Proprietary ONLY to NE Switch ]

### NATIONAL NETWORKS :

1. VISA 🡪 PIVISA
2. CIRRUS 🡪 PICIRD [ PIN ATM ]
3. MEASTRO 🡪 PICIRD [ PIN POS ]
4. MASTER CARD 🡪 PIMOLDD [ SIGNATURED ]

### REGIONAL NETWORKS :

1. PINYCED
2. PISHAZD
3. PIPULSED
4. PIACEXD 🡪 ACCEL
5. PIALLPD 🡪 ALL POINT
6. PIELAND 🡪 MONEY PASS
7. PICU24D 🡪 PIFIS

## **CDA (Consolidated Database Administration)** –

* It is used to CREATE, RETRIEVE and MODIFY Switch Database Records.
* It is also used to VERIFY the Information entered in Database Records and Compile that Data into the Configuration Files required to RUN the Switch.

### STEPS TO LOGON TO CDA –

1. At the TACL prompt type ->

>> CDA

1. Select F4 to choose PI94 Switch.
2. Select F1. It will have a PROCESSOR, CARD and TERMINAL Database.
3. If we want to Create a NEW Processor, we need to Click F2.

### CDA Function KEYS –

* F3 🡪 Help
* F5 🡪 READ the NEXT Record in the Database
* F6 🡪 READ the FIRST Record with a record key greater than or Equal to Current key.
* F10 🡪 ADD a Record
* F12 🡪 DELETE a Record
* F14 🡪 UPDATE a Record
* F16 🡪 RETURN to Previous Screen
* SF16 🡪 RETURN to the Main Database Screen

## **CODING** **& UNIT TESTING**–

### CODING 🡪

STEPS –

1. After the STORY is assigned, Check on which MODULE the change is Required.
2. Produce the LATEST MODULE from CONTROL, Duplicate the Module for Backup and make Changes on the Module in your Location.
3. After Making Changes – Do a “RENUM ALL, 1” and change the date and version in the Last Line of the Code.
4. COMPILE your Code 🡪 “TALS <File-name>”
5. BIND you Code 🡪 “BS <filename>”
6. Check for Errors in PERUSE 🡪 “PERUSE” >> “J <jobname>; LA”.

To clear all the Jobs from PERUSE 🡪 “SPD”

## **WEBFASTEST** / **PARAGON** -

1. Make sure that the ACQUIRER and ISSUER Processes are already Created and PROCESS has to be pointed to the LATEST OBJECT (present in your location).
2. Make sure you have a MACCARD / GATECARD created and setup in WEBFASTEST.
3. Make sure the ACQUIRER and ISSUER Process are UP and RUNNING.

# ***ENFORM***

## **How to WRITE ENFORM QUERY** –

STEPS ---

1. We First give the DICTIONARY in which the DATA are Stored.

Syntax :-

?DICTIONARY <dictionary-path>

e.g., ?DICTIONARY \NSCD2.$DAT036.EPSDICT

1. We then ASSIGN a Label (File Name) to the FILE we are searching for and give the File Path.

Syntax :-

?ASSIGN <filename> TO <file-path>

e.g., ?ASSIGN MACCARD TO \NSCD2.COMMEVDB.MACCARD

1. We then OPEN the FILE using the Assigned Name.

Syntax :-

OPEN <filename>

e.g., OPEN MACCARD

1. We then write the QUERY to search the Data.