ESB Implementation Architecture

### Mini Statement

|  |  |
| --- | --- |
| **Version** | **1.0** |
| **Created on** | **2o Apr,2020** |
| **Prepared by** | **Eidiko Systems Integrators** |
|  |  |
| Document Classification | CONFIDENTIAL |
| Distribution | Eidiko, Steward Bank |

Table of Contents

[MiniStatement 1](file:///C:\Users\bandaru\Downloads\Direct%20Debit%20Outward_V4.docx#_Toc472606341)

[Table of Contents 2](#_Toc472606342)

[1 Approval 4](#_Toc472606343)

[2 Revision History 4](#_Toc472606344)

[3 Introduction 5](#_Toc472606345)

[3.1 General Description 5](#_Toc472606346)

[4 Integration Overview 5](#_Toc472606347)

[4.1 Business Process 5](#_Toc472606348)

[4.1.1 Mini Statement 5](#_Toc472606349)

[4.2 High Level Architecture 8](#_Toc472606350)

[4.3 Detailed Architecture 8](#_Toc472606351)

[4.4 Dependencies 9](#_Toc472606352)

[4.4.1 External Dependencies 9](#_Toc472606353)

[4.4.2 Internal Dependencies 9](#_Toc472606354)

[4.4.3 Consumer 10](#_Toc472606355)

[4.4.4 Provider 12](#_Toc472606356)

[4.4.5 Mapping 28](#_Toc472606357)

[4.4.6 Error Handling 37](#_Toc472606358)

[4.4.7 Retry Mechanism 37](#_Toc472606359)

[4.4.8 Reference Data 37](#_Toc472606360)

[5 Database Definitions 37](#_Toc472606361)

[5.1 Scripts 37](#_Toc472606362)

[6 Queue Definitions 41](#_Toc472606363)

[6.1 Scripts 41](#_Toc472606364)

[7 Status Request from OAB 42](#_Toc472606365)

[7.1 Batch status 42](#_Toc472606366)

[7.2 Transaction status 43](#_Toc472606367)

[8 Integration Definition 45](#_Toc472606368)

[8.1 Direct Debit Outward 45](#_Toc472606369)

[8.1.1 IntegrationDetails 45](#_Toc472606370)

[9 Requirements 46](#_Toc472606371)

[9.1 Integration Requirements 46](#_Toc472606372)

[9.2 Business Requirements 47](#_Toc472606373)

[9.3 Non-functional Requirements 47](#_Toc472606374)

[9.3.1 Volume 47](#_Toc472606375)

[9.3.2 Performance 47](#_Toc472606376)

[9.3.3 Security 47](#_Toc472606377)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Role | Approver | Company | Email | Email |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Version | Author |
| **20 Apr, 2020** | Initial Version | 1.0 | P V Sattibabu |

# Approval

# Revision History

# Introduction

## General Description

A mini statement service is used to get the details from particular bank(steward bank) and it will give acknowledge to the particular customer.

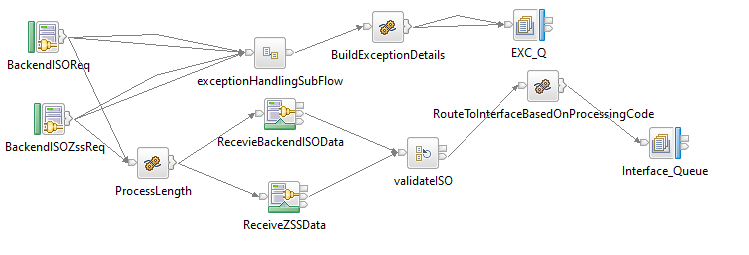
# 4. Integration Overview

## 4.1 Business Process

Mini statement request is initiated by a user/customer of steward bank to know the transaction details from an account.

### 4.1.1 Mini Statement

**Router Flow :IncomingRouterFlow.msgflow**

****

**CODE:**

1.In the “ProcessLength” Compute node we are calculating the tcpip header size.

CREATE COMPUTE MODULE IncomingRouterFlow\_ProcessLength

DECLARE LOG\_CONFIG\_PATH EXTERNAL CHARACTER '';

DECLARE LOG4J\_INIT\_ERROR\_MSG EXTERNAL CHARACTER '';

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

DECLARE rc BOOLEAN FALSE;

CALL initLog4j(LOG\_CONFIG\_PATH) INTO rc;

IF ( rc = FALSE ) THEN

DECLARE msg CHARACTER ;

SET Environment.Variables.Log4j.ErrorMsg = LOG4J\_INIT\_ERROR\_MSG;

END IF;

DECLARE ccidRef INTEGER InputRoot.Properties.CodedCharSetId;

DECLARE encodeRef INTEGER InputRoot.Properties.Encoding;

DECLARE inDomainDataRef REFERENCE TO InputRoot.DFDL;

DECLARE I INTEGER 0;

DECLARE J,AA INTEGER;

DECLARE sumAll INTEGER 0;

SET J = CARDINALITY(InputRoot.DFDL.ISO8583TCPHeader.\*[]) - 1;

DECLARE it INTEGER CARDINALITY(InputRoot.DFDL.ISO8583TCPHeader.\*[]);

WHILE I <= J DO

SET AA = InputRoot.DFDL.ISO8583TCPHeader.[I+1];

SET it = it - 1;

SET sumAll = sumAll + POWER(2,it) \* AA;

SET I = I + 1;

END WHILE;

SET OutputLocalEnvironment.TCPIP.Receive.Length = sumAll;

SET OutputLocalEnvironment.TCPIP.Receive.Id = InputLocalEnvironment.TCPIP.Input.ConnectionDetails.Id;

SET Environment.ReqLength = sumAll;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','..............Start logging header data...........') INTO rc;

SET OutputRoot.Properties = InputRoot.Properties;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','Calculated Length:'||CAST(sumAll AS CHARACTER)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','RAW Data:'||getPayLoad(FIELDNAME(InputBody),inDomainDataRef,encodeRef,ccidRef)) INTO rc;

DECLARE portRef CHARACTER CAST(InputLocalEnvironment.TCPIP.Input.ConnectionDetails.Port AS CHARACTER);

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','Incoming Port:'||portRef) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','Incoming conn id:'||CAST(InputLocalEnvironment.TCPIP.Input.ConnectionDetails.Id AS CHARACTER)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','server input TCP open timestamp:' ||CAST(InputLocalEnvironment.TCPIP.Input.ConnectionDetails.OpenTimestamp AS CHARACTER)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','server input TCP close timestamp:' ||CAST(InputLocalEnvironment.TCPIP.Input.ConnectionDetails.CloseTimestamp AS CHARACTER)) INTO rc;

IF CONTAINS(portRef,'8806') THEN

PROPAGATE TO TERMINAL 'out';

ELSE

PROPAGATE TO TERMINAL 'out1';

END IF;

CALL writeToLogFile(MessageFlowLabel, 'Log', 'DEBUG','..............end logging header data...........') INTO rc;

RETURN FALSE;

END;

END MODULE;

**CODE:**

1.in the “RouteToInterfaceBasedOnProcessingCode” Compute node we arefeatching the details like ministatement flow queue name(“MS\_SB\_TCP\_REQ”) and response queue name(“ALL\_SB\_TCP\_RES”) based on the procces code and placed in enviranment

CREATE COMPUTE MODULE IncomingRouterFlow\_RouteToInterfaceBasedonProcessingCode

DECLARE DSN EXTERNAL CHARACTER '';

DECLARE SCH EXTERNAL CHARACTER '';

DECLARE IsLogRequired EXTERNAL CHARACTER '';

DECLARE LOG\_CONFIG\_PATH EXTERNAL CHARACTER '';

DECLARE LOG4J\_INIT\_ERROR\_MSG EXTERNAL CHARACTER '';

DECLARE rc BOOLEAN;

DECLARE NETWORKPROCESSCODE EXTERNAL CHARACTER '';

DECLARE RAW\_AUDIT\_Q EXTERNAL CHARACTER '';

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

CALL CopyEntireMessage();

RETURN FALSE;

END;

CREATE PROCEDURE CopyEntireMessage() BEGIN

CALL initLog4j(LOG\_CONFIG\_PATH) INTO rc;

IF ( rc = FALSE ) THEN

DECLARE msg CHARACTER ;

SET Environment.Variables.Log4j.ErrorMsg = LOG4J\_INIT\_ERROR\_MSG;

END IF;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','===================================== INCOMMING ROUTER =================================') INTO rc ;

DECLARE ccidRef INTEGER InputRoot.Properties.CodedCharSetId;

DECLARE encodeRef INTEGER InputRoot.Properties.Encoding;

DECLARE domainDataRef REFERENCE TO InputRoot.DFDL;

DECLARE domainName CHARACTER FIELDNAME(InputBody);

SET Environment.Variables.reqMsg = getPayLoad(domainName,domainDataRef,encodeRef,ccidRef);

DECLARE inRef REFERENCE TO InputRoot.DFDL.ISO8583\_1987;

DECLARE version CHARACTER inRef.MTI\_Version;

DECLARE class CHARACTER inRef.MTI\_MessageClass;

DECLARE funct CHARACTER inRef.MTI\_MessageFunction;

DECLARE origin CHARACTER inRef.MTI\_MessageOrigin;

DECLARE procCodeData CHARACTER;

DECLARE DBDetails ROW;

DECLARE QueueName CHARACTER;

DECLARE refProcessingCode CHARACTER inRef.ProcessingCode\_003;

DECLARE tcpipDetails REFERENCE TO InputLocalEnvironment.TCPIP.Receive.ConnectionDetails;

DECLARE procCode,reqMsg CHARACTER InputRoot.DFDL.ISO8583\_1987.ProcessingCode\_003;

SET procCodeData = SUBSTRING(procCode FROM 1 FOR 2);

-- ==================== TCPIP DETAILS AND ISO8583 DETAILS =========================

Set OutputRoot.MQRFH2.usr.Type = tcpipDetails.Type;

Set OutputRoot.MQRFH2.usr.Hostname = tcpipDetails.Hostname;

Set OutputRoot.MQRFH2.usr.Port = tcpipDetails.Port;

Set OutputRoot.MQRFH2.usr.Id = tcpipDetails.Id;

Set OutputRoot.MQRFH2.usr.ReplyId = tcpipDetails.ReplyId;

Set OutputRoot.MQRFH2.usr.OpenTimestamp = tcpipDetails.OpenTimestamp;

Set OutputRoot.MQRFH2.usr.reqMsg = Environment.Variables.reqMsg;

Set OutputRoot.MQRFH2.usr.timeLocalTransaction = CAST(inRef.SystemsTraceAuditNumber\_011 AS CHARACTER);

Set OutputRoot.MQRFH2.usr.dateLocalTransaction = inRef.DateLocalTransaction\_013;

Set OutputRoot.MQRFH2.usr.retrievalReferenceNumber = inRef.RetrievalReferenceNumber\_037;

SET Environment.MQRFH2 = OutputRoot.MQRFH2;

SET Environment.Variables.Id = tcpipDetails.Id;

SET Environment.Variables.systemsTraceAuditNumber = CAST(inRef.SystemsTraceAuditNumber\_011 AS CHARACTER);

SET Environment.Variables.dateLocalTransaction = inRef.DateLocalTransaction\_013;

SET Environment.Variables.retrievalReferenceNumber = inRef.RetrievalReferenceNumber\_037;

DECLARE req CHARACTER;

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','ISO Request: '|| Environment.Variables.reqMsg) INTO rc ;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','TCPIP ID: '|| tcpipDetails.Id)INTO rc;

END IF;

--==========================START OF RAW AUDIT DB LOGGING ======================

SET Environment.Variables.msgID = CAST(Environment.MQRFH2.usr.Id AS CHARACTER );

DECLARE outRefer REFERENCE TO OutputRoot;

CALL DBLogging(Environment.Variables.msgID,Environment.Variables.reqMsg,'Router Request',ApplicationLabel,BrokerName,Environment.MQRFH2.usr.timeLocalTransaction,Environment.MQRFH2.usr.dateLocalTransaction,Environment.MQRFH2.usr.retrievalReferenceNumber,outRefer);

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = RAW\_AUDIT\_Q;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','===================================== Populating to RAW\_Audit\_Q =================================') INTO rc ;

PROPAGATE TO TERMINAL 'out' ;

--========================== END OF RAW AUDIT DB LOGGING ======================

SET OutputRoot.Properties= InputRoot.Properties;

SET OutputRoot.MQRFH2= Environment.MQRFH2;

IF FIELDNAME(inRef.ResponseCode\_039) IS NOT NULL AND (version||class||funct||origin = '0210' OR version||class||funct||origin = '0410') THEN

IF EXISTS(getDBDetailsCache.Values[]) THEN

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','Values available in shared cache: '||getDBDetailsCache.Values.\*[])INTO rc;

--SET DBDetails.values[] = select A.PROC\_CODE,A.REQ\_QUEUE\_NAME,A.RESQ\_QUEUE\_NAME,A.DOMAIN from Database.DSN.SYSTEM.INTEGRATION\_SOL\_DETAILS AS A WHERE A.PROC\_CODE = procCodeData and A.DOMAIN = domainName;

DECLARE qDetails ROW;

SET qDetails.values[] = SELECT P.HTTP\_REQ, P.HTTP\_RES ,P.HTTP\_BUSINESS\_TEMP FROM getDBDetailsCache.Values[] AS P WHERE P.PROC\_CODE = procCodeData;

SET QueueName = qDetails.values.HTTP\_BUSINESS\_TEMP;

SET OutputRoot.MQRFH2.usr.resQueueName = qDetails.values.HTTP\_RES;

ELSE

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','Calling the Cache')INTO rc;

CALL STORE\_CACHE(procCodeData);

DECLARE DBValues ROW;

SET DBValues.values[] = SELECT T.HTTP\_REQ, T.HTTP\_RES ,T.HTTP\_BUSINESS\_TEMP FROM getDBDetailsCache.Values[] AS T WHERE T.PROC\_CODE = procCodeData;

SET QueueName = DBValues.values.HTTP\_BUSINESS\_TEMP;

SET OutputRoot.MQRFH2.usr.resQueueName = DBValues.values.HTTP\_RES;

END IF;

SET OutputRoot.DFDL = InputRoot.DFDL;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = QueueName;

PROPAGATE TO TERMINAL 'out';

ELSE

--IF domainName LIKE 'DFDL' THEN

IF EXISTS(getDBDetailsCache.Values[]) THEN

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','Values available in shared cache: '||getDBDetailsCache.Values.\*[])INTO rc;

--SET DBDetails.values[] = select A.PROC\_CODE,A.REQ\_QUEUE\_NAME,A.RESQ\_QUEUE\_NAME,A.DOMAIN from Database.DSN.SYSTEM.INTEGRATION\_SOL\_DETAILS AS A WHERE A.PROC\_CODE = procCodeData and A.DOMAIN = domainName;

DECLARE qDetails ROW;

IF version||class||funct||origin = '0800' OR version||class||funct||origin = '0801' THEN

SET qDetails.values[] = SELECT P.REQ\_QUEUE, P.RES\_QUEUE FROM getDBDetailsCache.Values[] AS P WHERE P.PROC\_CODE = NETWORKPROCESSCODE;

ELSE

SET qDetails.values[] = SELECT P.REQ\_QUEUE, P.RES\_QUEUE FROM getDBDetailsCache.Values[] AS P WHERE P.PROC\_CODE = procCodeData;

END IF;

SET QueueName = qDetails.values.REQ\_QUEUE;

SET OutputRoot.MQRFH2.usr.resQueueName = qDetails.values.RES\_QUEUE;

ELSE

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','Calling the Cache')INTO rc;

CALL STORE\_CACHE(procCodeData);

DECLARE DBValues ROW;

IF version||class||funct||origin = '0800' OR version||class||funct||origin = '0801' THEN

SET DBValues.values[] = SELECT T.REQ\_QUEUE, T.RES\_QUEUE FROM getDBDetailsCache.Values[] AS T WHERE T.PROC\_CODE = NETWORKPROCESSCODE;

ELSE

SET DBValues.values[] = SELECT T.REQ\_QUEUE, T.RES\_QUEUE FROM getDBDetailsCache.Values[] AS T WHERE T.PROC\_CODE = procCodeData;

END IF;

SET QueueName = DBValues.values.REQ\_QUEUE;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','============Stored Cache================================='||CAST(getDBDetailsCache.Values AS CHARACTER)) INTO rc ;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG',DBValues.values.REQ\_QUEUE) INTO rc ;

SET OutputRoot.MQRFH2.usr.resQueueName = DBValues.values.RES\_QUEUE;

END IF;

SET OutputRoot.DFDL = InputRoot.DFDL;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','=====================================Business QUEUENAME=================================') INTO rc ;

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG',QueueName) INTO rc ;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = QueueName;

PROPAGATE TO TERMINAL 'out' ;

END IF;

END;

CREATE PROCEDURE STORE\_CACHE (IN procCodeData CHARACTER)

BEGIN

-- A.PROC\_CODE, A.REQ\_QUEUE, A.RES\_QUEUE

SET getDBDetailsCache.Values[] = SELECT A.PROC\_CODE,A.REQ\_QUEUE,A.RES\_QUEUE,A.HTTP\_REQ,A.HTTP\_RES,A.HTTP\_BUSINESS\_TEMP FROM Database.INTEGRATION\_SOL\_DETAILS AS A; --WHERE A.PROC\_CODE = procCodeData;

IF getDBDetailsCache.Values IS NOT NULL OR getDBDetailsCache.Values <> '' THEN

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','Retrieved values from Database')INTO rc;

ELSE

CALL writeToLogFile(MessageFlowLabel,'Log','DEBUG','No Data retrieved from Database')INTO rc;

END IF;

END;

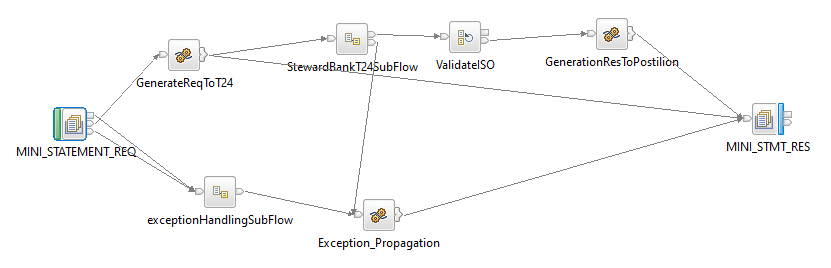
END MODULE;

Process Flow:

1. Postilion sends Mini statement Message from channel to App Connect Enterprise TCPIP Server Input.
2. The format of the request Message sent by Postilion is specified in Section 5.1.2.1
3. TCPIP Server Input Processes the incoming Mini Statement Message Request.
4. The Message is validated using RESET CONTENT DESCRIPTOR node and send to compute node
5. Compute node then filter queues based on Processing Code 38 gets queues MS\_SB\_TCP\_REQ and ALL\_SB\_TCP\_RES.
6. Compute Node Property i.e compute mode is set to LocalEnvironment and Meassage. Then set queue name dynamically using esql to the request queue i. MS\_SB\_TCP\_REQ, so that message is routed to Interface specific application.
7. In this flow Incoming Message is Logged in Database and file system(using Log4j).

**Img3.**

**ESB Flow :** **SB\_MiniStatement.msgflow**

****

**CODE:**

1.In the “GenerateReqToT24” Compute node we are storing the request in database logging and log4j logging and sending the request to the “T24”.

CREATE COMPUTE MODULE MiniStatementFlow\_GenerateReqToT24

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

CALL GenerateReqToT24();

RETURN TRUE;

END;

CREATE PROCEDURE GenerateReqToT24() BEGIN

SET Environment.MQRFH2 = InputRoot.MQRFH2;

DECLARE rc BOOLEAN FALSE;

CALL initLog4j(LOG\_CONFIG\_PATH) INTO rc;

IF ( rc = FALSE ) THEN

DECLARE msg CHARACTER ;

SET Environment.Variables.Log4j.ErrorMsg = LOG4J\_INIT\_ERROR\_MSG;

END IF;

DECLARE inRef REFERENCE TO InputRoot.DFDL.ISO8583\_1987;

DECLARE ccidRef INTEGER InputRoot.Properties.CodedCharSetId;

DECLARE encodeRef INTEGER InputRoot.Properties.Encoding;

DECLARE domainDataRef REFERENCE TO InputRoot.DFDL;

DECLARE domainName CHARACTER FIELDNAME(InputBody);

SET Environment.Variables.reqMsg = getPayLoad(domainName,domainDataRef,encodeRef,ccidRef);

DECLARE outRefer REFERENCE TO OutputRoot;

--------------DBLogging-------------------

IF IsLogRequired LIKE 'Y' THEN

CALL DBLogging(Environment.MQRFH2.usr.Id,Environment.Variables.reqMsg,'Application Request',ApplicationLabel,BrokerName,Environment.MQRFH2.usr.timeLocalTransaction,Environment.MQRFH2.usr.dateLocalTransaction,

Environment.MQRFH2.usr.retrievalReferenceNumber,outRefer);

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = RAW\_AUDIT\_Q;

PROPAGATE TO TERMINAL 'out1';

-----------------------------

DECLARE env REFERENCE TO Environment.Variables;

SET OutputRoot.Properties = InputRoot.Properties;

SET OutputRoot.MQMD = InputRoot.MQMD;

CREATE LASTCHILD OF OutputRoot DOMAIN ('DFDL');

DECLARE outRef REFERENCE TO OutputRoot.DFDL;

CREATE LASTCHILD OF OutputRoot.DFDL AS outRef NAME 'ISO8583\_1987';

--set response queue and tcpip connection id in environment

DECLARE resQ CHARACTER Environment.MQRFH2.usr.resQueueName;

DECLARE connId CHARACTER Environment.MQRFH2.usr.Id;

SET Environment.Variables.ResponseQ = resQ;

SET Environment.Variables.TcpIp.ConnectionId = connId;

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','..............Initializing Mini Statement logging...........') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','..............ISO Message parsed successfully...........') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','Request from Postilion::') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG',getPayLoad(domainName,inRef,encodeRef,ccidRef)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','TCPIP ID::'|| InputRoot.MQRFH2.usr.Id) INTO rc;

END IF;

--field to field mapping

CALL FieldToFieldMapping(inRef,outRef,'');

SET Environment.dfdlData = OutputRoot.DFDL;

----- Database Logging ------------

IF IsLogRequired LIKE 'Y' THEN

CALL DBLogging(Environment.MQRFH2.usr.Id,getPayLoad(domainName,outRef,encodeRef,ccidRef),'T24 Request',ApplicationLabel,BrokerName,Environment.MQRFH2.usr.timeLocalTransaction,Environment.MQRFH2.usr.dateLocalTransaction,Environment.MQRFH2.usr.retrievalReferenceNumber,outRefer);

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = RAW\_AUDIT\_Q;

SET OutputRoot.DFDL = NULL;

PROPAGATE TO TERMINAL 'out1';

----------------------

SET OutputRoot.Properties=InputRoot.Properties;

SET OutputRoot.MQRFH2 = Environment.MQRFH2;

SET OutputRoot.DFDL = Environment.dfdlData;

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','Request To T24::') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG',getPayLoad(domainName,outRef,encodeRef,ccidRef)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','TCPIP ID::'|| InputRoot.MQRFH2.usr.Id) INTO rc;

END IF;

END;

END MODULE;

**CODE:**

1.In the “GenerationResToPostilion” Computre node we are getting the response form “T24” and storeing the response in database logging and log4j logging and sending to the “outgoing router”.

CREATE COMPUTE MODULE MiniStatementFlow\_GenerateResToPostilion

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

CALL GenerateResToPostilion();

RETURN FALSE;

END;

CREATE PROCEDURE GenerateResToPostilion() BEGIN

DECLARE rc BOOLEAN FALSE;

CALL initLog4j(LOG\_CONFIG\_PATH) INTO rc;

IF ( rc = FALSE ) THEN

DECLARE msg CHARACTER ;

SET Environment.Variables.Log4j.ErrorMsg = LOG4J\_INIT\_ERROR\_MSG;

END IF;

SET Environment.Properties = InputRoot.Properties;

SET Environment.MQMD = InputRoot.MQMD;

DECLARE encodeRef REFERENCE TO InputRoot.Properties.Encoding;

DECLARE ccidRef REFERENCE TO InputRoot.Properties.CodedCharSetId;

DECLARE domainName CHARACTER FIELDNAME(InputBody);

DECLARE domainDataRef REFERENCE TO InputRoot.DFDL;

DECLARE inRef REFERENCE TO InputRoot.DFDL.ISO8583\_1987;

DECLARE outRefer REFERENCE TO OutputRoot;

SET Environment.Variables.resData = getPayLoad(domainName,domainDataRef,encodeRef,ccidRef);

-----------DBLogging------------

IF IsLogRequired LIKE 'Y' THEN

CALL DBLogging(Environment.MQRFH2.usr.Id,Environment.Variables.resData,'T24 Response',ApplicationLabel,BrokerName,inRef.SystemsTraceAuditNumber\_011,inRef.DateLocalTransaction\_013,inRef.RetrievalReferenceNumber\_037,outRefer);

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = RAW\_AUDIT\_Q;

SET Environment.resData = OutputRoot.JSON.Data;

PROPAGATE TO TERMINAL 'out';

--------------------

CREATE LASTCHILD OF OutputRoot DOMAIN ('DFDL');

DECLARE outRef REFERENCE TO OutputRoot.DFDL;

CREATE LASTCHILD OF OutputRoot.DFDL AS outRef NAME 'ISO8583\_1987';

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','..........Logging generated Response...............') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','T24 Response::') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG',getPayLoad(domainName,inRef,encodeRef,ccidRef)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','TCPIP ID::'|| Environment.MQRFH2.usr.Id) INTO rc;

END IF;

--field to field mapping

CALL FieldToFieldMapping(inRef,outRef,'');

SET Environment.dfdlData = OutputRoot.DFDL;

----- Database Logging ------------

IF IsLogRequired LIKE 'Y' THEN

CALL DBLogging(Environment.MQRFH2.usr.Id,getPayLoad(domainName,outRef,encodeRef,ccidRef),'Application Response',ApplicationLabel,BrokerName,Environment.MQRFH2.usr.timeLocalTransaction,Environment.MQRFH2.usr.dateLocalTransaction,Environment.MQRFH2.usr.retrievalReferenceNumber,outRefer);

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = RAW\_AUDIT\_Q;

SET OutputRoot.DFDL = NULL;

PROPAGATE TO TERMINAL 'out';

----------------------

SET OutputRoot.Properties = Environment.Properties;

SET OutputRoot.MQRFH2 = Environment.MQRFH2;

SET OutputRoot.MQMD = InputRoot.MQMD;

SET OutputRoot.DFDL = Environment.dfdlData;

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','Mini Statement Response To Postilion::') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG',getPayLoad(domainName,outRef,encodeRef,ccidRef)) INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'minidebug', 'DEBUG','TCPIP ID::'|| Environment.MQRFH2.usr.Id) INTO rc;

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = Environment.MQRFH2.usr.resQueueName;

PROPAGATE TO TERMINAL 'out';

END;

END MODULE;

**CODE:**

1.In the “Exception\_Propagation” Compute node we are capturing the exception and creating user define exception message and storing the data in to the datanase and log4j logging.

CREATE COMPUTE MODULE MiniStatementFlow\_CaptureException

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

CALL BuildErrorMsg();

RETURN FALSE;

END;

CREATE PROCEDURE BuildErrorMsg() BEGIN

DECLARE rc BOOLEAN FALSE;

CALL initLog4j(LOG\_CONFIG\_PATH) INTO rc;

IF ( rc = FALSE ) THEN

SET Environment.Variables.Log4j.ErrorMsg = LOG4J\_INIT\_ERROR\_MSG;

END IF;

DECLARE excpRef REFERENCE TO InputRoot.XMLNSC.ExceptionDetails;

DECLARE outRefer REFERENCE TO OutputRoot;

DECLARE encodeRef REFERENCE TO InputRoot.Properties.Encoding;

DECLARE ccidRef REFERENCE TO InputRoot.Properties.CodedCharSetId;

DECLARE domainName CHARACTER FIELDNAME(InputBody);

--DECLARE Id CHARACTER CAST(Environment.MQRFH2.usr.Id AS CHARACTER);

DECLARE inRef REFERENCE TO InputRoot.XMLNSC;

CREATE LASTCHILD OF OutputRoot DOMAIN 'XMLNSC';

CREATE LASTCHILD OF OutputRoot.XMLNSC NAME 'Ministatement';

DECLARE outRef REFERENCE TO OutputRoot.XMLNSC.\*;

CALL BuildExceptionDetails(excpRef,outRef,'Ministatement');

SET Environment.Variables.UserDefinedErrorCodes = OutputRoot.XMLNSC;

DECLARE domainDataRef REFERENCE TO OutputRoot.XMLNSC;

DECLARE exe\_Desc CHARACTER getPayLoad(domainName,domainDataRef,encodeRef,ccidRef);

IF IsLogRequired LIKE 'Y' THEN

CALL writeToLogFile(MessageFlowLabel, 'ErrorLogger', 'DEBUG','..............Logging Exception ...........') INTO rc;

CALL writeToLogFile(MessageFlowLabel, 'ErrorLogger', 'ERROR','Application Built Exception:'||getPayLoad(domainName,domainDataRef,encodeRef,ccidRef)) INTO rc;

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = EXCEPTION\_Q;

PROPAGATE TO TERMINAL 'out';

------------Error Logging in DB----------

IF IsLogRequired LIKE 'Y' THEN

CALL Error\_DBLogging(Environment.MQRFH2.usr.Id,Environment.Variables.reqMsg,'Application Error',ApplicationLabel,BrokerName,exe\_Desc,Environment.MQRFH2.usr.timeLocalTransaction,Environment.MQRFH2.usr.dateLocalTransaction,Environment.MQRFH2.usr.retrievalReferenceNumber,CAST(excpRef.excpNumber AS CHARACTER),Environment.Variables.UserDefinedErrorCodes.Ministatement.ErrorCode,outRefer);

CALL writeToLogFile(MessageFlowLabel, 'ErrorLogger', 'ERROR','Exception Created:'||getPayLoad(domainName,domainDataRef,encodeRef,ccidRef)) INTO rc;

END IF;

SET OutputLocalEnvironment.Destination.MQ.DestinationData[1].queueName = ERR\_AUDIT\_Q;

PROPAGATE TO TERMINAL 'out';

----------------------------------

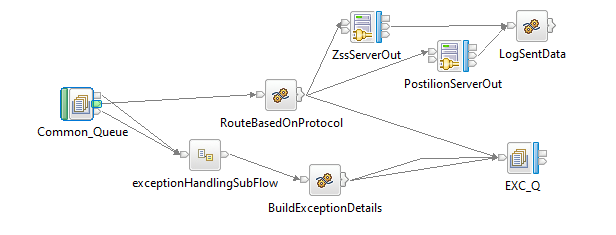
END;

END MODULE;

Process Flow:

1. The message is routed from the IncomingRouterFlow to the interface ESB application through MS\_SB\_TCP\_REQ queue in MQ\_InputNode(MINI\_STATEMENT\_REQ).
2. The format of the Mini Statement Send is DB Logged and the data is logged using Log4j is specified in Section 5.1.2.1.
3. On unsuccessful processing the exception is DB Logged and mentioned into the queue.
4. And the request to T24 Temenos is generated in the compute node GeneratingReqToT24.
5. The StewardBankT24 call will be processed here on the basis of processing code.
6. The ResetContentDescriptor is used to validate the T24 Response message.
7. Node GeneratingResToPostilion, the T24 response message will be placed in the Database by DB Logging and inserted into the log file(MINISTATEMENT.log) of Log4j.
8. On unsuccessful processing the exception is mentioned into the exception queue and the ExceptionHandler will handle the esceptions.
9. On successful processing of the T24 response the message is propagated to the response queue (ALL\_SB\_TCP\_RES).
10. From the ESB message flow (**Mini Statement Flow**) it will propagate to OutwardRouter queue which is picked from the Environment variables
11. All the queue names here are either taken from the Environment variables and by setting them in User Defined Properties.

**Router Flow :OutgoingRouterFlow.msgflow**

****

Process Flow:

1. The incoming message from the Mini Statement Flow to the response queue ALL\_SB\_TCP\_RES.
2. In OutgoingRouterFlow the Compute node RouteBasedOnProtocol is used to route according to the incoming message.
3. TCPIP Server Output Processes the Mini Statement Message Response.
4. On unsuccessful processing the exception is logged into the EXC\_Q queue.
5. The format of the Mini Statement message response will be in section
6. The TCPIP server output of App Connect Enterprise will send the Mini statement response back to the Postilion.

## 4.2 High Level Architecture

Following is the high level architecture diagram.

1

Channel

6

2

Postilion

3

5

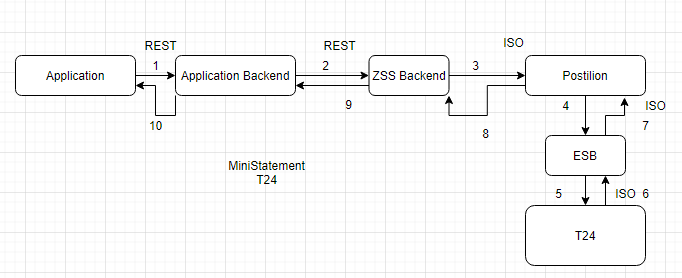
ESB

4

Temenos(T24)

## Detailed Architecture

Following is the detailed architecture diagram.



## Dependencies

### 4.4.1 External Dependencies

The following table lists external dependencies in the ESB solution perspective.

|  |  |
| --- | --- |
| # | External Dependency |
|  | Postilion Systems |
|  | Teminos(T24) System |

### 4.4.2 Internal Dependencies

The following table lists internal dependencies in the ESB solution perspective.

|  |  |
| --- | --- |
| # | Internal Dependency |
|  | MQ Server |
|  | DB Server |

### 

### 4.4.3 Consumer

#### 4.4.3.1 Request from Ministatement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | DFDL Element | Data Type | Mandatory | Length of the Field |
|  | PrimaryAccountNumber\_002 | LL | M | **1-99** |
| 2 | ProcessingCode\_003 | String | M | 6 |
| 3 | AmountTransaction\_004 | Decimal | M | 12 |
| 4 | AmountSettlement\_005 | Decimal | O | 12 |
| 5 | AmountCardHolderBilling\_006 | Decimal | O | 12 |
| 6 | TransmissionDatetime\_007 | dateTiemNoYear | M | 10 |
| 7 | AmountCardHolderBillingFee\_008 | Decimal | O | 8 |
| 8 | ConversionRateSettlement\_009 | String |  | 8 |
| 9 | ConversionRateCardholderBilling\_010 | String | O | 8 |
| 10 | SystemsTraceAuditNumber\_011 | String | M | 6 |
| 11 | TimeLocalTransaction\_012 | Time | M | **6** |
| 12 | DateLocalTransaction\_013 | monthDay | M | 4 |
| 13 | DateExpiration\_014 | yearMonth | M | 4 |
| 14 | DateSettlement\_015 | monthDay | M | 4 |
| 15 | DateConversion\_016 | monthDay | O | 4 |
| 16 | DateCapture\_017 | monthDay | O | 4 |
| 17 | MerchantType\_018 | String | M | 4 |
| 18 | AcquiringInstitutionCountryCode\_019 | String | O | 3 |
| 19 | PANExtendedCountryCode\_020 | String | O | 3 |
| 20 | ForwardingInstitutionCountryCode\_021 | String | O | 3 |
| 21 | PointOfServiceEntryMode\_022 | String | M | 3 |
| 22 | CardSequenceNumber\_023 | String | O | 3 |
| 23 | NetworkInternationalIdentifier\_024 | String | O | 3 |
| 24 | PointOfServiceConditionCode\_025 | String | M | 2 |
| 25 | PointOfServiceCaptureCode\_026 | String | O | 2 |
| 26 | AuthorizingIdentificationResponseLength\_027 | String |  | 1 |
| 27 | AmountTransactionFee\_028 | String | M | 9 |
| 28 | AmountSettlementFee\_029 | String | O | 9 |
| 29 | AmountTransactionProcessingFee\_030 | String | M | 9 |
| 30 | AmountSettlementProcessingFee\_031 | String | O | 9 |
| 31 | AcquiringInstitutionIdentificationCode\_032 | LL | M | 1-99 |
| 32 | ForwardingInstitutionIdentificationCode\_033 | LL | M | 1-99 |
| 33 | PrimaryAccountNumberExtended\_034 | LL | O | 1-99 |
| 34 | Track2Data\_035 | LL | O | 1-99 |
| 35 | Track3Data\_036 | LLL | O | 1-999 |
| 36 | RetrievalReferenceNumber\_037 | String | M | 12 |
| 37 | AuthorizationIdentificationResponse\_038 | String | O | 6 |
| 38 | ResponseCode\_039 | String | O | 2 |
| 39 | ServiceRestrictionCode\_040 | String | O | 3 |
| 40 | CardAcceptorTerminalIdentification\_041 | String | M | 8 |
| 41 | CardAcceptorIdentificationCode\_042 | String | M | 15 |
| 42 | CardAcceptorNameLocation\_043 | String | M | 40 |
| 43 | AdditionalResponseData\_044 | LL | O | 1-99 |
| 44 | Track1Data\_045 | LL | O | 1-99 |
| 45 | AdditionalDataISO\_046 | LLL | O | 1-999 |
| 46 | AdditionalDataNational\_047 | LLL | O | 1-999 |
| 47 | AdditionalDataPrivate\_048 | LLL | O | 1-999 |
| 48 | CurrencyCodeTransaction\_049 | String | M | 3 |
| 49 | CurrencyCodeSettlement\_050 | String | O | 3 |
| 50 | CurrencyCodeCardholderBilling\_051 | String | O | 3 |
| 51 | PersonalIdentificationNumberData\_052 | String | O | 8 |
| 52 | SecurityRelatedControlInformation\_053 | String | O | 16 |
| 53 | AdditionalAmounts\_054 | LLL | O | 1-999 |
| 54 | ReservedISO\_055 | LLL | O | 1-999 |
| 55 | ReservedISO\_056 | LLL | M | 1-999 |
| 56 | ReservedNational\_057 | LLL | O | 1-999 |
| 57 | ReservedNational\_058 | LLL | O | 1-999 |
| 58 | ReservedNational\_059 | LLL | M | 1-999 |
| 59 | AdviceReasonCode\_060 | LLL | O | 1-999 |
| 60 | ReservedPrivate\_061 | LLL | O | 1-999 |
| 61 | ReservedPrivate\_062 | LLL | O | 1-999 |
| 62 | ReservedPrivate\_063 | LLL | O | 1-999 |
| 63 | MessageAuthenticationCode\_064 | hexBinary | O | 8 |
| 64 | TertiaryBitmap\_065 | hexBinary | O | 8 |
| 65 | SettlementCode\_066 | String | O | 1 |
| 66 | ExtendedPaymentCode\_067 | String | O | 2 |
| 67 | ReceivingInstitutionCountryCode\_068 | String | O | 3 |
| 68 | SettlementInstitutionCountryCode\_069 | String | O | 3 |
| 69 | NetworkManagementInformationCode\_070 | String | O | 3 |
| 70 | MessageNumber\_071 | String | O | 4 |
| 71 | MessageNumberLast\_072 | String | O | 4 |
| 72 | DateAction\_073 | date | O | 6 |
| 73 | CreditsNumber\_074 | String | O | 10 |
| 74 | CreditsReversalNumber\_075 | String | O | 10 |
| 75 | DebitsNumber\_076 | String | O | 10 |
| 76 | DebitsReversalNumber\_077 | String | O | 10 |
| 77 | TransferNumber\_078 | String | O | 10 |
| 78 | TransferReversalNumber\_079 | String | O | 10 |
| 79 | InquiriesNumber\_080 | String | O | 10 |
| 80 | AuthorizationsNumber\_081 | String | O | 10 |
| 81 | CreditsProcessingFeeAmount\_082 | String | O | 12 |
| 82 | CreditsTransactionFeeAmount\_083 | String | O | 12 |
| 83 | DebitsProcessingFeeAmount\_084 | String | O | 12 |
| 84 | DebitsTransactionFeeAmount\_085 | String | O | 12 |
| 85 | CreditsAmount\_086 | String | O | 16 |
| 86 | CreditsReversalAmount\_087 | String | O | 16 |
| 87 | DebitsAmount\_088 | String | O | 16 |
| 88 | DebitsReversalAmount\_089 | String | O | 16 |
| 89 | OriginalDataElements\_090 | String | O | 42 |
| 90 | FileUpdateCode\_091 | String | O | 1 |
| 91 | FileSecurityCode\_092 | String | O | 2 |
| 92 | ResponseIndicator\_093 | String | O | 5 |
| 93 | ServiceIndicator\_094 | String | O | 7 |
| 94 | ReplacementAmounts\_095 | String | O | 42 |
| 95 | MessageSecurityCode\_096 | String | O | 8 |
| 96 | AmountNetSettlement\_097 | String | O | 16 |
| 97 | Payee\_098 | String | O | 25 |
| 98 | SettlementInstitutionIdentificationCode\_099 | LL | O | 1-99 |
| 99 | ReceivingInstitutionIdentificationCode\_100 | LL | M | 1-99 |
| 100 | FileName\_101 | LL | O | 1-99 |
| 101 | AccountIdentification1\_102 | LL | M | 1-99 |
| 102 | AccountIdentification2\_103 | LL | O | 1-99 |
| 103 | TransactionDescription\_104 | LL | O | 1-99 |
| 104 | ReservedISO\_105 | LLL | O | 1-999 |
| 105 | ReservedISO\_106 | LLL | O | 1-999 |
| 106 | ReservedISO\_107 | LLL | O | 1-999 |
| 107 | ReservedISO\_108 | LLL | O | 1-999 |
| 108 | ReservedISO\_109 | LLL | O | 1-999 |
| 109 | ReservedISO\_110 | LLL | O | 1-999 |
| 110 | ReservedISO\_111 | LLL | O | 1-999 |
| 111 | ReservedNational\_112 | LLL | O | 1-999 |
| 112 | ReservedNational\_113 | LLL | O | 1-999 |
| 113 | ReservedNational\_114 | LLL | O | 1-999 |
| 114 | ReservedNational\_115 | LLL | O | 1-999 |
| 115 | ReservedNational\_116 | LLL | O | 1-999 |
| 116 | ReservedNational\_117 | LLL | O | 1-999 |
| 117 | ReservedNational\_118 | LLL | O | 1-999 |
| 118 | ReservedNational\_119 | LLL | O | 1-999 |
| 119 | ReservedPrivate\_120 | LLL | O | 1-999 |
| 120 | ReservedPrivate\_121 | LLL | O | 1-999 |
| 121 | ReservedPrivate\_122 | LLL | O | 1-999 |
| 122 | ReservedPrivate\_123 | LLL | M | 1-999 |
| 123 | ReservedPrivate\_124 | LLL | O | 1-999 |
| 124 | ReservedPrivate\_125 | LLL | O | 1-999 |
| 125 | ReservedPrivate\_126 | LLL | O | 1-999 |
| 126 | ReservedPrivate\_127 | String | M | 6 |
| 127 | Postilion\_Bitmaps\_Group\_127\_1 |  | O |  |
| 128 | SwitchKeyPostilionPrivate\_127\_2 | LL | M | 1-99 |
| 129 | RoutingInformationPostilionPrivate\_127\_3 | String | M | 48 |
| 130 | POSDataPostilionPrivate\_127\_4 | String | O | 22 |
| 131 | ServiceStationDataPostilionPrivate\_127\_5 | String | O | 73 |
| 132 | AuthorizationProfilePostilionPrivate\_127\_6 | String | O | 2 |
| 133 | CheckDataPostilionPrivate\_127\_7 | LL | O | 1-99 |
| 134 | RetentionDataPostilionPrivate\_127\_8 | LLL | O | 1-999 |
| 135 | AdditionalNodeDataPostilionPrivate\_127\_9 | LLL | O | 1-999 |
| 136 | CVVPostilionPrivate\_127\_10 | String | O | 3 |
| 137 | OriginalKeyPostilionPrivate\_127\_11 | LL | O | 1-99 |
| 138 | OriginalKeyPostilionPrivate\_127\_12 | LL | O | 1-99 |
| 139 | PosGeographicDataPostilionPrivate\_127\_13 | String | O | 17 |
| 140 | SponsorBankPostilionPrivate\_127\_14 | String | O | 8 |
| 141 | AddressVerificationDataPostilionPrivate\_127\_15 | LL | O | 1-99 |
| 142 | AddressVerificationResultPostilionPrivate\_127\_16 | String | O | 1 |
| 143 | CardHolderInformationPostilionPrivate\_127\_17 | LL | O | 1-99 |
| 144 | ValidationDataPostilionPrivate\_127\_18 | LL | O | 1-99 |
| 145 | BankDetailsPostilionPrivate\_127\_19 | String | O | 31 |
| 146 | OriginatorAuthorizerDatePostilionPrivate\_127\_20 | String | M | 8 |
| 147 | RecordIdentificationPostilionPrivate\_127\_21 | LL | O | 1-99 |
| 148 | StructuredDataPostilionPrivate\_127\_22 | LLLLL | M | 1-99999 |
| 149 | PayeeNameAddressPostilionPrivate\_127\_23 | String | O | 253 |
| 150 | PayerAccountPostilionPrivate\_127\_24 | LL | O | 1-99 |
| 151 | ICCDataPostilionPrivate\_127\_25 | LLLL | O | 1-9999 |
| 152 | ICCData\_Bitmaps\_Group\_127\_25\_1 |  | O |  |
|  | AmountAuthorizedPostilionPrivate\_127\_25\_2 | String | O | 12 |
| 153 | AmountOtherPostilionPrivate\_127\_25\_3 | String | O | 12 |
| 154 | ApplicationIdentifierPostilionPrivate\_127\_25\_4 | LL | O | 1-99 |
| 155 | ApplicationInterchangeProfilePostilionPrivate\_127\_25\_5 | String | O | 4 |
| 156 | ApplicationTransactionCounterPostilionPrivate\_127\_25\_6 | String | O | 4 |
| 157 | ApplicationUsageControlPostilionPrivate\_127\_25\_7 | String | O | 4 |
| 158 | AuthorizationResponseCodePostilionPrivate\_127\_25\_8 | String | O | 2 |
| 159 | CardAuthReliabilityIndicatiorPostilionPrivate\_127\_25\_9 | String | O | 1 |
| 160 | CardAuthResultsCodePostilionPrivate\_127\_25\_10 | String | O | 1 |
| 161 | ChipConditionCodePostilionPrivate\_127\_25\_11 | String | O | 1 |
| 162 | CryptogramPostilionPrivate\_127\_25\_12 | String | O | 16 |
| 163 | CryptogramInformationDataPostilionPrivate\_127\_25\_13 | String | O | 2 |
| 164 | CvmListPostilionPrivate\_127\_25\_14 | LLL | O | 1-999 |
| 165 | CvmResultsPostilionPrivate\_127\_25\_15 | String | O | 6 |
| 166 | InterfaceDeviceSerialNumberPostilionPrivate\_127\_25\_16 | String | O | 8 |
| 167 | IssuerActionCodePostilionPrivate\_127\_25\_17 | String | O | 11 |
| 168 | IssuerApplicationDataPostilionPrivate\_127\_25\_18 | LL | O | 1-99 |
| 169 | IssuerScriptResultsPostilionPrivate\_127\_25\_19 | LLL | O | 1-999 |
| 170 | TerminalApplicationVersionNumberPostilionPrivate\_127\_25\_20 | String | O | 4 |
| 171 | TerminalCapabilitiesPostilionPrivate\_127\_25\_21 | String | O | 6 |
| 172 | TerminalCountryCodePostilionPrivate\_127\_25\_22 | String | O | 3 |
| 173 | TerminalTypePostilionPrivate\_127\_25\_23 | String | O | 2 |
| 174 | TerminalVerificationResultPostilionPrivate\_127\_25\_24 | String | O | 10 |
| 175 | TerminalCategoryCodePostilionPrivate\_127\_25\_25 | String | O | 1 |
| 176 | TransactionCurrencyCodePostilionPrivate\_127\_25\_26 | String | O | 3 |
| 177 | TransactionDatePostilionPrivate\_127\_25\_27 | String | O | 6 |
| 178 | TransactionSequenceCounterPostilionPrivate\_127\_25\_28 | L | O | 1-9 |
| 179 | TransactionTypePostilionPrivate\_127\_25\_29 | String | O | 2 |
| 180 | UnpredictableNumberResultPostilionPrivate\_127\_25\_30 | String | O | 8 |
| 181 | IssuerAuthDataPostilionPrivate\_127\_25\_31 | LL | O | 1-99 |
| 182 | IssuerScriptTemplate1PostilionPrivate\_127\_25\_32 | LLLL | O | 1-9999 |
| 183 | IssuerScriptTemplate2PostilionPrivate\_127\_25\_33 | LLLL | O | 1-9999 |
| 184 | OriginalNodePostilionPrivate\_127\_26 | LL | O | 1-99 |
| 185 | CardVerificationResultPostilionPrivate\_127\_27 | String | O | 1 |
| 186 | AmericanExpressCardIdentifierPostilionPrivate\_127\_28 | String | O | 4 |
| 187 | SecureData3DPostilionPrivate\_127\_29 | hexBinary | O | 40 |
| 188 | SecureData3DResultPostilionPrivate\_127\_30 | String | O | 1 |
| 189 | IssuerNetworkIDPostilionPrivate\_127\_31 | LL | O | 1-99 |
| 190 | UCAFDataPostilionPrivate\_127\_32 | LL | O | 1-99 |
| 191 | ExtendedTransactionTypePostilionPrivate\_127\_33 | String | O | 4 |
| 192 | AccountTypeQualifierPostilionPrivate\_127\_34 | String | O | 2 |
| 193 | AcquirerNetworkIDPostilionPrivate\_127\_35 | LL | O | 1-99 |
| 194 | CustomerIDPostilionPrivate\_127\_36 | LL | O | 1-99 |
| 195 | ExtendedResponseCodePostilionPrivate\_127\_37 | String | O | 4 |
| 195 | AdditionalPOSDataCodePostilionPrivate\_127\_38 | LL | O | 1-99 |
| 196 | OriginalResponseCodePostilionPrivate\_127\_39 | String | O | 2 |
| 197 | MessageAuthenticationCode\_128 | hexBinary | O | 8 |

#### 4.4.3.2 Response

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | DFDL Element | Data Type | Mandatory | Length of the Field |
|  | PrimaryAccountNumber\_002 | LL | M | **1-99** |
| 2 | ProcessingCode\_003 | String | M | 6 |
| 3 | AmountTransaction\_004 | Decimal | M | 12 |
| 4 | AmountSettlement\_005 | Decimal | O | 12 |
| 5 | AmountCardHolderBilling\_006 | Decimal | O | 12 |
| 6 | TransmissionDatetime\_007 | dateTiemNoYear | M | 10 |
| 7 | AmountCardHolderBillingFee\_008 | Decimal | O | 8 |
| 8 | ConversionRateSettlement\_009 | String | O | 8 |
| 9 | ConversionRateCardholderBilling\_010 | String | O | 8 |
| 10 | SystemsTraceAuditNumber\_011 | String | M | 6 |
| 11 | TimeLocalTransaction\_012 | Time | M | **6** |
| 12 | DateLocalTransaction\_013 | monthDay | M | 4 |
| 13 | DateExpiration\_014 | yearMonth | M | 4 |
| 14 | DateSettlement\_015 | monthDay | M | 4 |
| 15 | DateConversion\_016 | monthDay | O | 4 |
| 16 | DateCapture\_017 | monthDay | O | 4 |
| 17 | MerchantType\_018 | String | M | 4 |
| 18 | AcquiringInstitutionCountryCode\_019 | String | O | 3 |
| 19 | PANExtendedCountryCode\_020 | String | O | 3 |
| 20 | ForwardingInstitutionCountryCode\_021 | String | O | 3 |
| 21 | PointOfServiceEntryMode\_022 | String | M | 3 |
| 22 | CardSequenceNumber\_023 | String | O | 3 |
| 23 | NetworkInternationalIdentifier\_024 | String | O | 3 |
| 24 | PointOfServiceConditionCode\_025 | String | M | 2 |
| 25 | PointOfServiceCaptureCode\_026 | String | O | 2 |
| 26 | AuthorizingIdentificationResponseLength\_027 | String | O | 1 |
| 27 | AmountTransactionFee\_028 | String | M | 9 |
| 28 | AmountSettlementFee\_029 | String | O | 9 |
| 29 | AmountTransactionProcessingFee\_030 | String | M | 9 |
| 30 | AmountSettlementProcessingFee\_031 | String | O | 9 |
| 31 | AcquiringInstitutionIdentificationCode\_032 | LL | M | 1-99 |
| 32 | ForwardingInstitutionIdentificationCode\_033 | LL | M | 1-99 |
| 33 | PrimaryAccountNumberExtended\_034 | LL | O | 1-99 |
| 34 | Track2Data\_035 | LL | O | 1-99 |
| 35 | Track3Data\_036 | LLL | O | 1-999 |
| 36 | RetrievalReferenceNumber\_037 | String | M | 12 |
| 37 | AuthorizationIdentificationResponse\_038 | String | O | 6 |
| 38 | ResponseCode\_039 | String | M | 2 |
| 39 | ServiceRestrictionCode\_040 | String | O | 3 |
| 40 | CardAcceptorTerminalIdentification\_041 | String | M | 8 |
| 41 | CardAcceptorIdentificationCode\_042 | String | M | 15 |
| 42 | CardAcceptorNameLocation\_043 | String | M | 40 |
| 43 | AdditionalResponseData\_044 | LL | O | 1-99 |
| 44 | Track1Data\_045 | LL | O | 1-99 |
| 45 | AdditionalDataISO\_046 | LLL | O | 1-999 |
| 46 | AdditionalDataNational\_047 | LLL | O | 1-999 |
| 47 | AdditionalDataPrivate\_048 | LLL | M | 1-999 |
| 48 | CurrencyCodeTransaction\_049 | String | M | 3 |
| 49 | CurrencyCodeSettlement\_050 | String | O | 3 |
| 50 | CurrencyCodeCardholderBilling\_051 | String | O | 3 |
| 51 | PersonalIdentificationNumberData\_052 | String | O | 8 |
| 52 | SecurityRelatedControlInformation\_053 | String | O | 16 |
| 53 | AdditionalAmounts\_054 | LLL | M | 1-999 |
| 54 | ReservedISO\_055 | LLL | O | 1-999 |
| 55 | ReservedISO\_056 | LLL | M | 1-999 |
| 56 | ReservedNational\_057 | LLL | O | 1-999 |
| 57 | ReservedNational\_058 | LLL | O | 1-999 |
| 58 | ReservedNational\_059 | LLL | M | 1-999 |
| 59 | AdviceReasonCode\_060 | LLL | O | 1-999 |
| 60 | ReservedPrivate\_061 | LLL | O | 1-999 |
| 61 | ReservedPrivate\_062 | LLL | O | 1-999 |
| 62 | ReservedPrivate\_063 | LLL | O | 1-999 |
| 63 | MessageAuthenticationCode\_064 | hexBinary | O | 8 |
| 64 | TertiaryBitmap\_065 | hexBinary | O | 8 |
| 65 | SettlementCode\_066 | String | O | 1 |
| 66 | ExtendedPaymentCode\_067 | String | O | 2 |
| 67 | ReceivingInstitutionCountryCode\_068 | String | O | 3 |
| 68 | SettlementInstitutionCountryCode\_069 | String | O | 3 |
| 69 | NetworkManagementInformationCode\_070 | String | O | 3 |
| 70 | MessageNumber\_071 | String | O | 4 |
| 71 | MessageNumberLast\_072 | String | O | 4 |
| 72 | DateAction\_073 | date | O | 6 |
| 73 | CreditsNumber\_074 | String | O | 10 |
| 74 | CreditsReversalNumber\_075 | String | O | 10 |
| 75 | DebitsNumber\_076 | String | O | 10 |
| 76 | DebitsReversalNumber\_077 | String | O | 10 |
| 77 | TransferNumber\_078 | String | O | 10 |
| 78 | TransferReversalNumber\_079 | String | O | 10 |
| 79 | InquiriesNumber\_080 | String | O | 10 |
| 80 | AuthorizationsNumber\_081 | String | O | 10 |
| 81 | CreditsProcessingFeeAmount\_082 | String | O | 12 |
| 82 | CreditsTransactionFeeAmount\_083 | String | O | 12 |
| 83 | DebitsProcessingFeeAmount\_084 | String | O | 12 |
| 84 | DebitsTransactionFeeAmount\_085 | String | O | 12 |
| 85 | CreditsAmount\_086 | String | O | 16 |
| 86 | CreditsReversalAmount\_087 | String | O | 16 |
| 87 | DebitsAmount\_088 | String | O | 16 |
| 88 | DebitsReversalAmount\_089 | String | O | 16 |
| 89 | OriginalDataElements\_090 | String | O | 42 |
| 90 | FileUpdateCode\_091 | String | O | 1 |
| 91 | FileSecurityCode\_092 | String | O | 2 |
| 92 | ResponseIndicator\_093 | String | O | 5 |
| 93 | ServiceIndicator\_094 | String | O | 7 |
| 94 | ReplacementAmounts\_095 | String | O | 42 |
| 95 | MessageSecurityCode\_096 | String | O | 8 |
| 96 | AmountNetSettlement\_097 | String | O | 16 |
| 97 | Payee\_098 | String | O | 25 |
| 98 | SettlementInstitutionIdentificationCode\_099 | LL | O | 1-99 |
| 99 | ReceivingInstitutionIdentificationCode\_100 | LL | M | 1-99 |
| 100 | FileName\_101 | LL | O | 1-99 |
| 101 | AccountIdentification1\_102 | LL | M | 1-99 |
| 102 | AccountIdentification2\_103 | LL | O | 1-99 |
| 103 | TransactionDescription\_104 | LL | O | 1-99 |
| 104 | ReservedISO\_105 | LLL | O | 1-999 |
| 105 | ReservedISO\_106 | LLL | O | 1-999 |
| 106 | ReservedISO\_107 | LLL | O | 1-999 |
| 107 | ReservedISO\_108 | LLL | O | 1-999 |
| 108 | ReservedISO\_109 | LLL | O | 1-999 |
| 109 | ReservedISO\_110 | LLL | O | 1-999 |
| 110 | ReservedISO\_111 | LLL | O | 1-999 |
| 111 | ReservedNational\_112 | LLL | O | 1-999 |
| 112 | ReservedNational\_113 | LLL | O | 1-999 |
| 113 | ReservedNational\_114 | LLL | O | 1-999 |
| 114 | ReservedNational\_115 | LLL | O | 1-999 |
| 115 | ReservedNational\_116 | LLL | O | 1-999 |
| 116 | ReservedNational\_117 | LLL | O | 1-999 |
| 117 | ReservedNational\_118 | LLL | O | 1-999 |
| 118 | ReservedNational\_119 | LLL | O | 1-999 |
| 119 | ReservedPrivate\_120 | LLL | O | 1-999 |
| 120 | ReservedPrivate\_121 | LLL | O | 1-999 |
| 121 | ReservedPrivate\_122 | LLL | O | 1-999 |
| 122 | ReservedPrivate\_123 | LLL | M | 1-999 |
| 123 | ReservedPrivate\_124 | LLL | O | 1-999 |
| 124 | ReservedPrivate\_125 | LLL | O | 1-999 |
| 125 | ReservedPrivate\_126 | LLL | O | 1-999 |
| 126 | ReservedPrivate\_127 | String | M | 6 |
| 127 | Postilion\_Bitmaps\_Group\_127\_1 |  | O |  |
| 128 | SwitchKeyPostilionPrivate\_127\_2 | LL | M | 1-99 |
| 129 | RoutingInformationPostilionPrivate\_127\_3 | String | M | 48 |
| 130 | POSDataPostilionPrivate\_127\_4 | String | O | 22 |
| 131 | ServiceStationDataPostilionPrivate\_127\_5 | String | O | 73 |
| 132 | AuthorizationProfilePostilionPrivate\_127\_6 | String | O | 2 |
| 133 | CheckDataPostilionPrivate\_127\_7 | LL | O | 1-99 |
| 134 | RetentionDataPostilionPrivate\_127\_8 | LLL | O | 1-999 |
| 135 | AdditionalNodeDataPostilionPrivate\_127\_9 | LLL | O | 1-999 |
| 136 | CVVPostilionPrivate\_127\_10 | String | O | 3 |
| 137 | OriginalKeyPostilionPrivate\_127\_11 | LL | O | 1-99 |
| 138 | OriginalKeyPostilionPrivate\_127\_12 | LL | O | 1-99 |
| 139 | PosGeographicDataPostilionPrivate\_127\_13 | String | O | 17 |
| 140 | SponsorBankPostilionPrivate\_127\_14 | String | O | 8 |
| 141 | AddressVerificationDataPostilionPrivate\_127\_15 | LL | O | 1-99 |
| 142 | AddressVerificationResultPostilionPrivate\_127\_16 | String | O | 1 |
| 143 | CardHolderInformationPostilionPrivate\_127\_17 | LL | O | 1-99 |
| 144 | ValidationDataPostilionPrivate\_127\_18 | LL | O | 1-99 |
| 145 | BankDetailsPostilionPrivate\_127\_19 | String | O | 31 |
| 146 | OriginatorAuthorizerDatePostilionPrivate\_127\_20 | String | M | 8 |
| 147 | RecordIdentificationPostilionPrivate\_127\_21 | LL | O | 1-99 |
| 148 | StructuredDataPostilionPrivate\_127\_22 | LLLLL | M | 1-99999 |
| 149 | PayeeNameAddressPostilionPrivate\_127\_23 | String | O | 253 |
| 150 | PayerAccountPostilionPrivate\_127\_24 | LL | O | 1-99 |
| 151 | ICCDataPostilionPrivate\_127\_25 | LLLL | O | 1-9999 |
| 152 | ICCData\_Bitmaps\_Group\_127\_25\_1 |  | O |  |
|  | AmountAuthorizedPostilionPrivate\_127\_25\_2 | String | O | 12 |
| 153 | AmountOtherPostilionPrivate\_127\_25\_3 | String | O | 12 |
| 154 | ApplicationIdentifierPostilionPrivate\_127\_25\_4 | LL | O | 1-99 |
| 155 | ApplicationInterchangeProfilePostilionPrivate\_127\_25\_5 | String | O | 4 |
| 156 | ApplicationTransactionCounterPostilionPrivate\_127\_25\_6 | String | O | 4 |
| 157 | ApplicationUsageControlPostilionPrivate\_127\_25\_7 | String | O | 4 |
| 158 | AuthorizationResponseCodePostilionPrivate\_127\_25\_8 | String | O | 2 |
| 159 | CardAuthReliabilityIndicatiorPostilionPrivate\_127\_25\_9 | String | O | 1 |
| 160 | CardAuthResultsCodePostilionPrivate\_127\_25\_10 | String | O | 1 |
| 161 | ChipConditionCodePostilionPrivate\_127\_25\_11 | String | O | 1 |
| 162 | CryptogramPostilionPrivate\_127\_25\_12 | String | O | 16 |
| 163 | CryptogramInformationDataPostilionPrivate\_127\_25\_13 | String | O | 2 |
| 164 | CvmListPostilionPrivate\_127\_25\_14 | LLL | O | 1-999 |
| 165 | CvmResultsPostilionPrivate\_127\_25\_15 | String | O | 6 |
| 166 | InterfaceDeviceSerialNumberPostilionPrivate\_127\_25\_16 | String | O | 8 |
| 167 | IssuerActionCodePostilionPrivate\_127\_25\_17 | String | O | 11 |
| 168 | IssuerApplicationDataPostilionPrivate\_127\_25\_18 | LL | O | 1-99 |
| 169 | IssuerScriptResultsPostilionPrivate\_127\_25\_19 | LLL | O | 1-999 |
| 170 | TerminalApplicationVersionNumberPostilionPrivate\_127\_25\_20 | String | O | 4 |
| 171 | TerminalCapabilitiesPostilionPrivate\_127\_25\_21 | String | O | 6 |
| 172 | TerminalCountryCodePostilionPrivate\_127\_25\_22 | String | O | 3 |
| 173 | TerminalTypePostilionPrivate\_127\_25\_23 | String | O | 2 |
| 174 | TerminalVerificationResultPostilionPrivate\_127\_25\_24 | String | O | 10 |
| 175 | TerminalCategoryCodePostilionPrivate\_127\_25\_25 | String | O | 1 |
| 176 | TransactionCurrencyCodePostilionPrivate\_127\_25\_26 | String | O | 3 |
| 177 | TransactionDatePostilionPrivate\_127\_25\_27 | String | O | 6 |
| 178 | TransactionSequenceCounterPostilionPrivate\_127\_25\_28 | L | O | 1-9 |
| 179 | TransactionTypePostilionPrivate\_127\_25\_29 | String | O | 2 |
| 180 | UnpredictableNumberResultPostilionPrivate\_127\_25\_30 | String | O | 8 |
| 181 | IssuerAuthDataPostilionPrivate\_127\_25\_31 | LL | O | 1-99 |
| 182 | IssuerScriptTemplate1PostilionPrivate\_127\_25\_32 | LLLL | O | 1-9999 |
| 183 | IssuerScriptTemplate2PostilionPrivate\_127\_25\_33 | LLLL | O | 1-9999 |
| 184 | OriginalNodePostilionPrivate\_127\_26 | LL | O | 1-99 |
| 185 | CardVerificationResultPostilionPrivate\_127\_27 | String | O | 1 |
| 186 | AmericanExpressCardIdentifierPostilionPrivate\_127\_28 | String | O | 4 |
| 187 | SecureData3DPostilionPrivate\_127\_29 | hexBinary | O | 40 |
| 188 | SecureData3DResultPostilionPrivate\_127\_30 | String | O | 1 |
| 189 | IssuerNetworkIDPostilionPrivate\_127\_31 | LL | O | 1-99 |
| 190 | UCAFDataPostilionPrivate\_127\_32 | LL | O | 1-99 |
| 191 | ExtendedTransactionTypePostilionPrivate\_127\_33 | String | O | 4 |
| 192 | AccountTypeQualifierPostilionPrivate\_127\_34 | String | O | 2 |
| 193 | AcquirerNetworkIDPostilionPrivate\_127\_35 | LL | O | 1-99 |
| 194 | CustomerIDPostilionPrivate\_127\_36 | LL | O | 1-99 |
| 195 | ExtendedResponseCodePostilionPrivate\_127\_37 | String | O | 4 |
| 195 | AdditionalPOSDataCodePostilionPrivate\_127\_38 | LL | O | 1-99 |
| 196 | OriginalResponseCodePostilionPrivate\_127\_39 | String | O | 2 |
| 197 | MessageAuthenticationCode\_128 | hexBinary | O | 8 |

### 4.4.4 Error Handling

A Common flow by name "StewardBank\_Exception\_Handler" is designed so as to receive any exception and parse it. Upon parsing a corresponding database entry is made against the particular transaction. Subsequent to that an error log is updated with error description and id with the interface specific name.

### Retry Mechanism

|  |  |
| --- | --- |
| **Nr of retries** | **Comments/Details** |
| No of Retries | 3 in Configurable. |
| Time gap between retries | 1. Configurable. |

### Reference Data

|  |  |
| --- | --- |
| **Domain** | **Reference Data Table** |
| N/A | N/A |

# Database Definitions

**5.1 Scripts**

|  |  |  |  |
| --- | --- | --- | --- |
| **QM Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| AP\_SB\_TCP\_REQ | Receives the Messages from Channel via Router to ESB Flow. | <Final Script Will be placed, after complete Unit Testing> |
| ALL\_SB\_TCP\_RES | This is the Queue used to send to response back to the channel via router. |
| RAW\_LOG\_SB\_REQ | This queue is used to DB Log the successful transactions into the database. |
|  | ERR\_LOG\_SB\_REQ | This queue is used to DB Log the error/unsuccessful transactions into the database. |  |
|  | SB\_EXCQ | The common flow ‘StewardBankExceptionHandling’ uses this queue to store the exception details. |  |
|  | EXC\_Q | This queue is used to store the exceptions occurred in the flows. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TableName** | **Sl.No** | **Column Name** | **Data**  **Type** | **Purpose** |
| **INTEGRATION\_SOL\_DETAILS** | **1** | PROC\_CODE | VARCHAR2(20 BYTE) | Contains the unique number for each message details. Checks with the ISO message field ProcessingCode\_003 |
| **2** | REQ\_QUEUE | VARCHAR2(20 BYTE) | Contains the queue name of the Request queue from Mini Statement Flow. |
| **3** | RES\_QUEUE | VARCHAR2(20 BYTE) | Contains the Response queue name. |
| **4** | DOMAIN | VARCHAR2(20 BYTE) | Contains the Domain name of the message. |
| **TableName** | **Sl.No** | **Column Name** | **Data**  **Type** | **Purpose** |
| **RAW\_AUDIT\_LOGGING** | **1** | MSGID | VARCHAR2(100 BYTE) | Contains the unique number for each transaction details. |
| **2** | LOGGING\_TIME | TIMESTAMP(6) | Contains the timestamp, to know when the message was received. |
| **3** | MESSAGE | CLOB | Contains the received message. |
| **4** | MESSAGETYPE | VARCHAR2(100 BYTE | Contains the type of message is received(T24 Request, Application Request, Application Response, T24 Response). |
| **5** | APPNAME | VARCHAR2(100 BYTE | Contains the name of the application. |
| **6** | BROKER | VARCHAR2(100 BYTE | Contains the name of the broker. |
| **TableName** | **Sl.No** | **Column Name** | **Data**  **Type** | **Purpose** |
| VARCHAR2(8 BYTE) | **1** | MSGID | VARCHAR2(100 BYTE) | Contains the unique number for each transaction details. |
| **2** | LOGGING\_TIME | TIMESTAMP(6) | Contains the timestamp, to know when the message was received. |
| **3** | MESSAGE | CLOB | Contains the received message. |
| **4** | MESSAGETYPE | VARCHAR2(100 BYTE | Contains the type of message is received(T24 Request, Application Request, Application Response, T24 Response). |
| **5** | APPNAME | VARCHAR2(100 BYTE | Contains the name of the application. |
| **6** | BROKER | VARCHAR2(100 BYTE | Contains the name of the broker. |
| **7** | ERRORDESCRIPTION | CLOB | Contains the details of the error and its description. |

# Queue Definitions

## 6.1 Scripts

|  |  |  |  |
| --- | --- | --- | --- |
| **QM Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| MS\_SB\_TCP\_REQ | Receives the Messages from Channel via Router to ESB Mini Statement Flow. | <Final Script Will be placed, after complete Unit Testing> |
| ALL\_SB\_TCP\_RES | This is the Queue used to send to response back to the channel via router. |
| RAW\_LOG\_SB\_REQ | This queue is used to DB Log the successful transactions into the database. |
|  | ERR\_LOG\_SB\_REQ | Thisqueue is used to DB Log the error/unsuccessful transactions into the database. |  |
|  | SB\_EXCQ | The common flow ‘StewardBankExceptionHandling’ uses this queue to store the exception details. |  |
|  | EXC\_Q | This queue is used to store the exceptions occurred in the flows. |  |

# 

# Integration Definition

### IntegrationDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | **CBO** | |
| **Protocol/Format** | ISO8583/WMQ/DFDL | | ISO8583/WMQ/DFDL | |
| **Mode** | Async | | Async | |
| **Endpoint Info** | localhost:8806 | | localhost:8806 | |
| **Samples Messages**  **(XML / CSV**  **/ PDL /etc)** | XML Files | | XML Files | |
| POSTILION Request |  | Request to T24 |  |
| Outward Response to POSTILION |  | Response from T24 |  |
| **Resources**  **(XSD)** | POSTILION Request |  | Request to T24 |  | |
| Outward Response to POSTILION |  | Response from T24 |  | |

# Requirements

## Integration Requirements

This interface will perform these levels of validation:

|  |  |  |
| --- | --- | --- |
| # | Integration Requirement | Comments / Details |
| 1 | Schema Validation Requirements | Inherent with XSDs and message model of ISO8583\_1987. |
| 2 | Records Validation Requirements | According to the requirements. |
| 3 | Transformation Requirements | There are no transformations, the message is from ISO to ISO through out the flow. |
| 4 | Routing Requirements | Handled by the IncomingRouter and OutgoingRouter message flows. |
| 5 | Enrichment Requirements | According to the requirements. |
| 6 | Logging Requirements | Database Centric |
| 7 | Exception Handling Requirements | See “Error Handling” section above. |
| 8 | Retry Mechanism | Configurable |

## Business Requirements

|  |  |  |
| --- | --- | --- |
| # | Business Requirement | Comments / Details |
|  | N/A |  |

## Non-functional Requirements

The following are service specific non-functional requirements. Non-functional requirements already provided by the overall solution, as described in the Technical Design document, should not be referred here.

### Volume

|  |  |  |
| --- | --- | --- |
| # | Requirement | Detail |
| 1 | Records / Transaction / Files |  |
| 2 | Message Load size |  |

### Performance

|  |  |  |
| --- | --- | --- |
| # | Requirement | Detail |
| 1 | Frequency | Daily |

### Security

|  |  |  |
| --- | --- | --- |
| # | Requirement | Detail |
| 1 | Message security | All outgoing messages are digitally signed and sent. |
| 2 | Authentication | None |
| 3 | Authorization | None |