Network Messages Specification Low Level Design Document

**EIDIKO SYSTEMS INTEGRATORS**



Version: 1.0

Release Date: 07-09-2020

|  |  |  |
| --- | --- | --- |
| Prepared By | Reviewed By | Approved By |
| EIDIKO |  |  |

Document Classification: CONFIDENTIAL

Contents

[1 Introduction 5](#_Toc50412045)

[1.1 General Description 5](#_Toc50412046)

[2 Purpose 5](#_Toc50412047)

[3 Scope 5](#_Toc50412048)

[3.1 In Scope 5](#_Toc50412049)

[4 Interface Dependencies 5](#_Toc50412050)

[4.1 External Dependencies 5](#_Toc50412051)

[4.2 Internal Dependencies 6](#_Toc50412052)

[5 IIB Implementation Process Flow 6](#_Toc50412053)

[5.1 Process Diagram 6](#_Toc50412054)

[5.2 Process Flow Steps 6](#_Toc50412055)

[5.3 T24 Process workflow 7](#_Toc50412056)

[5.3.1 Process Flow Diagram 7](#_Toc50412057)

[5.3.2 Process Flow steps 7](#_Toc50412058)

[5.4 RAW\_AUDIT Process flow 8](#_Toc50412059)

[5.4.1 Process Flow Diagram 8](#_Toc50412060)

[5.4.2 Process Flow Steps: 8](#_Toc50412061)

[5.4 TCPIP Router Process Flow 9](#_Toc50412062)

[5.4.1 Incoming Router Process Flow Diagram: 9](#_Toc50412063)

[5.4.2 Process Flow Steps: 9](#_Toc50412064)

[5.4.3 STWB\_ESB\_TRAN\_DETAILS 10](#_Toc50412065)

[5.4.4 Outgoing Router Process Flow Diagram 11](#_Toc50412066)

[5.4.5 Process Flow Steps: 11](#_Toc50412067)

[5.5 Error Audit Process Details: 12](#_Toc50412068)

[5.5.1 Process Flow Diagram: 12](#_Toc50412069)

[5.5.2 Process Flow Steps: 13](#_Toc50412070)

[6 Interface Definitions 13](#_Toc50412071)

[6.1 Request Message Definition 13](#_Toc50412072)

[6.1.1 Channel Request Message Structure /Schema 13](#_Toc50412073)

[6.1.2 T24 Request Message Structure /Schema 13](#_Toc50412074)

[6.1.3 Data Base Response Codes 13](#_Toc50412075)

[6.2 Response Message Definition 14](#_Toc50412076)

[6.2.1 T24 & Channel Response Message Structure 14](#_Toc50412077)

[Response Type: ISO8583 14](#_Toc50412078)

[6.3 Integration Definition 14](#_Toc50412079)

[6.3.1 Integration Details 14](#_Toc50412080)

[7 Queue Details 15](#_Toc50412081)

[7.1 Table for Queue Details 15](#_Toc50412082)

[8 Logging Mechanism 16](#_Toc50412083)

[8.1 Insert into Raw\_Audit\_Table 16](#_Toc50412084)

[8.2 Insert into Error\_Audit\_Table 17](#_Toc50412085)

[8.3 File Based Logging 18](#_Toc50412086)

[8.3.1 Log4j 18](#_Toc50412087)

**Approval**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Role | Approver | Company | E-mail | Sign-Off Date |
|  |  | Eidiko |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Version | Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## General Description

A network management transaction is used to synchronize cryptographic keys between Postilion and the third party. A notification message is used to notify a third party, called a Control Node, of a switched transaction that took place between a source and sink node interface.

# Purpose

The purpose of this document is to capture events that trigger the interface, main steps within the interface and the integration architecture. This document is intended for use by the developers of the applications identified, the integration development team, and by the test organizations responsible for the testing of these applications.

# Scope

## In Scope

This document focuses on outlining the interface design for the Network Message ***(Request and Response)*** interface. Central to this document are the following:

* Overview of the business process that drives the need for the interface
* Proposed integration approach

# Interface Dependencies

## External Dependencies

The following table lists interface specific External requirements.

|  |  |
| --- | --- |
| **Item #** | **External Requirements** |
| 1 | MS SQL Database - 2017 |
| 2 | Core Banking System (Temenos-T24) |
| 3 | Postilion System |

## 

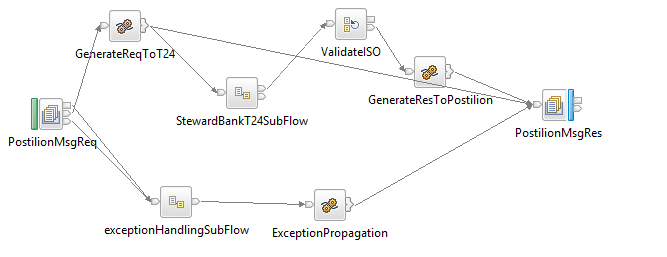
## Internal Dependencies

The following table lists interface specific internal requirements.

|  |  |
| --- | --- |
| **Item #** | **Internal Requirements** |
| 1 | TCPIP Router Interface (Gateway) - StewardBankTCPIPRouter |
| 2 | StewardBankT24Call |
| 3 | ISO8583\_Lib (Shared Library) |
| 4 | StewardBankCommonEsql\_Lib (Shared Library) |
| 5 | DB Logging App (Audit Logging) |
| 6 | StewardBank Exception Handler (Subflow) |

# IIB Implementation Process Flow

## Process Diagram



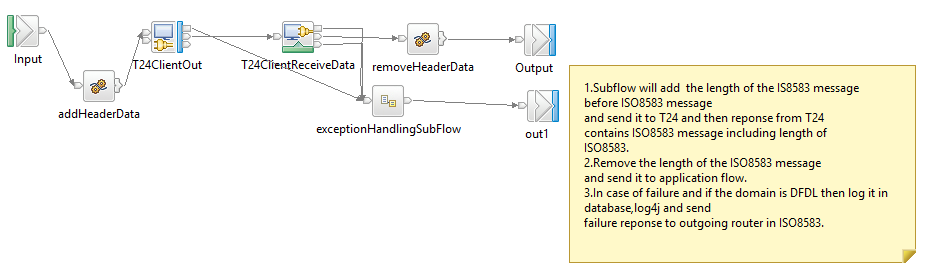
**Figure: 1 NetworkMessage Flow**

## Process Flow Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1. | ESB receives the request from the channel through the TCPIP Router Application. If the MTI message having 0800 or 0801 then form Database it fetches Queue(SB\_NETWK\_TCP\_REQ) and trigs the Business App(StewardBankPostilionMegsApp). |
| 2 | Initial request and response are logged in the RAW\_LOG\_SB\_REQ and inserting in Database (RAW\_AUDIT\_LOGGING). |
| 3 | After logging of request ESB frames the Request to T24 server i.e T24 server. |
| 4 | Then the request send to T24 .Then T24 sends the response for the network messages with the MTI 0810 . |
| 5 | After the Response is generated by T24 call same response is sent back to Channel (Postilion) through the TCPIP Router Application (OutgoingRouterFlow). |
| 6 | T24 Response is logged in RAW\_LOG\_SB\_REQ and is insert in Database (RAW\_AUDIT\_LOGGING). |
| 7 | For Unsuccesful Call error logs in ERROR\_LOG\_SB\_REQ and is inserted in Database(ERROR\_AUDIT\_LOGGING) |

## T24 Process workflow

### Process Flow Diagram



**Figure: 3 TCPIP T24 call for NetworkMessagesFlow (T24 Call)**

### Process Flow steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1 | The TCPIP Subflow will add the length of the IS8583 message before ISO8583 message and send it to T24 and then response from T24 contains ISO8583 message including length of ISO8583. |
| 2 | The Subflow will remove the length of the ISO8583 message.  And send it to ESB application flow. |
| 3 | In case of failure and if the domain is DFDL then log it in database,log4j and send failure response to outgoing router in ISO8583. |
| 4 | On Successful call of backend service, response will be generated and send back to Business Application for further process. |
| 5 | On Unsuccessful call of backend service it Retry for 3 times and sends error message to Business Application for further Process. |

## RAW\_AUDIT Process flow

### Process Flow Diagram



**Figure: 4 Raw\_Audit\_Logging Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The RAW\_AUDIT logs the input Request and Response |
|  | After the query is executed the record is inserted into the Database. |
|  | Final Response is inserted into the Database and then Response is logged into the RAW\_LOG\_SB\_REQ. |

## TCPIP Router Process Flow

### Incoming Router Process Flow Diagram:

****

**Figure: 5 Incoming Router Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | IIB receives the request from channel. |
|  | ESB receives the request from channel. |
|  | Request channel is the TCPIP request accepts the request from the channel |
|  | The request message is placed in the RAW\_LOG\_SB\_REQ to log the request in Database (RAW\_AUDIT\_LOGGING) |
|  | If the MTI is 8000 or 8001 then respective Queue is picked from STWB\_ESB\_TRAN\_DETAILS table and propagates it to the business flow (StewardBankPostilionMsgsApp). |
|  | The channel request (ISO8583) consists of header part, primary & secondary bitmaps and their fields along with the 127field subfields. Which is validated and header part is removed in the router and sent to the application. |
|  | It validates the processingCode and domain of input from STWB\_ESB\_TRAN\_DETAILS if it fails to validate the error response is logged in the logs and in Data Base (ERROR\_AUDIT\_LOG) |
|  | After the validation Request message is passed through ESB. If the queue name presents in MQ manager the ESB business flow triggers otherwise the error response is sent to the front end as Unable to open queue. |
|  | If it is success or failure response will store them in RAW\_AUDIT\_LOG table or ERROR\_AUDIT\_LOG and logging into log file. |
|  | The request from channel is sent to ESB the application. |

### STWB\_ESB\_TRAN\_DETAILS

**Table Name:** STWB\_ESB\_TRAN\_DETAILS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | APPLICATION\_NAME | Transaction Name | Varchar | 50 | N | This will indicate transaction name |
| 2 | SERVICE\_NAME | Service Name | Varchar | 30 | N | This will indicate transaction name |
| 3 | PROC\_CODE | Processing Code | Varchar | 20 | Y | From input request |
| 4 | TCP\_REQ\_QNAME | TCPIP Request Queue | Varchar | 20 | N | This Request Queue Name field is used for ISO Related Application |
| 5 | TCP\_RES\_QNAME | TCPIP Response Queue | Varchar | 20 | N | This Response Queue Name field is used for ISO Related Application |
| 6 | HTTP\_REQ\_QNAME | HTTP Request Queue | Varchar | 20 | Y | This Request Queue Name field is used for REST Related Application |
| 7 | HTTP\_RES\_QNAME | HTTP Response Queue | Varchar | 20 | Y | This Response Queue Name field is used for REST Related Application |
| 8 | CHANNEL | Channel | Varchar | 20 | Y | Channel Name |
| 9 | MSGDOMAIN | Message Domain | Varchar | 20 | Y | Request Message Format |

### Outgoing Router Process Flow Diagram



**Figure: 6 Outgoing Router Flow**

### Process Flow Steps:

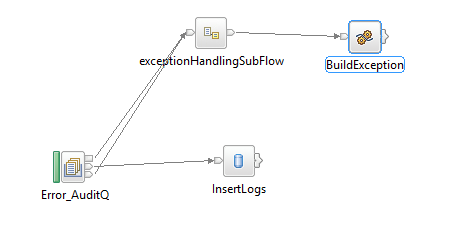
|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The OutgoingRouterFlow brings the Channel Response from ESB application from common response (ALL\_SB\_TCP\_RES)queue. |
|  | Channel Response will be logged in RAW\_LOG\_SB\_REQ and insert in Data Base. |
|  | Finally Channel Response will be send to the Channel(Postilion). |

## Error Audit Process Details:

### Process Flow Diagram:



**Figure: 7 Exception Subflow**



**Figure: 8 Error\_Audit\_Logging Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The ERROR\_AUDIT\_Q locks the input Request. |
|  | After the query is executed the record is inserted into the Data Base. |
|  | Final Response is inserted into the Database and then Response is logged into the ERR\_LOG\_SB\_REQ. |

# Interface Definitions

## Request Message Definition

### Channel Request Message Structure /Schema

**Request Type:** TCPIP

**Request Format:** ISO8583

### T24 Request Message Structure /Schema

**Request Type:** TCPIP

**Request Format:** ISO8583

### Data Base Response Codes

001 = "Fatal Exception";

002 = "Recoverable Exception";

003 = "Configuration Exception";

004 = "Security Exception";

005 = "Parser Exception";

006 = "Conversion Exception";

007 = "Data Base Exception";

008 = "User Exception";

009 = "Cast Exception";

010 = "Message Exception";

011 = "SQL Exception";

012 = "Socket Exception";

013 = "Socket Timeout Exception";

014 = "Unknown Exception";

015 = "Failure";

## Response Message Definition

### T24 & Channel Response Message Structure

### Response Type: ISO8583

## Integration Definition

### Integration Details

|  |  |  |
| --- | --- | --- |
| **Item #** | **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 throughout 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. |

# Queue Details

## Table for Queue Details

|  |  |  |  |
| --- | --- | --- | --- |
| **Queue Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| SB\_NETWK\_TCP\_REQ | Receives the Messages from Channel via Router to ESB NetworkMsgs 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. |  |

# Logging Mechanism

## Insert into Raw\_Audit\_Table

**Table Name: RAW\_AUDIT\_TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | MSGID | MessageID | Varchar | 100 | Y | From input request |
| 2 | LOGGING\_TIME | Logging Time | TimeStamp | 6 | Y | Name |
| 3 | MESSAGE | Message(Request/Response) | Clob | - | Y | Request from channel |
| 4 | MESSAGETYPE | Type of Message(Request/Response) | Varchar | 100 | Y | Type of message either Request or Response |
| 5 | APPNAME | Application Name | Varchar | 100 | Y | Name of the Application |
| 6 | BROKER | Broker Name | Varchar | 100 | Y | Broker Name |
| 7 | TIME\_LOCAL\_TRANSACTION | Transaction Time | Varchar | 20 | Y | Time of the Transaction |
| 8 | DATE\_LOCAL\_TRANSACTION | Transaction Date | Varchar | 20 | Y | Date of the Transaction |
| 9 | RETRIEVAL\_REFERENCE\_N | Retrieval Reference Number | Varchar | 20 | Y |  |

## Insert into Error\_Audit\_Table

**Table Name: ERROR\_AUDIT\_TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | MSGID | MessageID | Varchar | 50 | Y | From input request |
| 2 | LOGGING\_TIME | Application Name | Timestamp | 50 | Y | Time of the logging |
| 3 | MESSAGE | Message Type(Request/Response) | CLOB | 50 | Y | Request of the Message |
| 4 | MESSAGETYPE | Message Type | Varchar | 50 | Y | Type of message either Request or Response |
| 5 | APPNAME | Application Name | Varchar | 4000 | Y | Name of the Application |
| 6 | BROKER | Broker Name | Varchar | - | Y | Name of the Broker |
| 7 | ERRORDESCRIPTION | Exception | Clob | 4000 | Y | Exception Information |
|  | TIME\_LOCAL\_TRANSACTION | Transaction Time | Varchar | 20 | Y | Time of the Transaction |
|  | DATE\_LOCAL\_TRANSACTION | Transaction Date | Varchar | 20 | Y | Date of the Transaction |
|  | RETRIEVAL\_REFERENCE\_NUMBER | Retrieval Reference Number | Varchar | 20 | Y | Retrieval Reference Number |
|  | BORKER\_ERROR\_CODE | Broker Error Code | Varchar |  | Y | Error code generate by Broker |
|  | USER\_DEFINE\_ERROR\_CODE | User Defined Error Code | Varchar |  | Y | User Defined Code |

## File Based Logging

### Log4j

As part of auditing the request we have to use Log4j mechanism in the application to log the request in the file.

File Path: To be specified, while deploying the application.